FATS/FATAR/FATSCOPY USER'S MANUAL VERSION 4.9.28

FATS/FATAR/FATSCOPY is a package of tape utilities that includes:

- FATS (Fast Analysis of Tape Surfaces) provides certification and verification of tape volumes, erasing tapes and labeling tapes. FATS can be licensed separately.
- FATAR (Fast Analysis of Tape and Recovery) analyzes the contents of tape volumes and optionally copies the data to new tape volumes, including the ability to recover from errors on the input tapes, modify data during the copy, and rename and recatalog the output tape data sets. FATAR can optionally be called by FATS.
- FATSCOPY uses the copying facilities of FATAR to automate stacking of data sets on tape to improve utilization of high-capacity tape volumes and simplify the migration to new media types. FATSCOPY is an optional component of FATS/FATAR.

FATS, FATAR and FATSCOPY are proprietary products of

INNOVATION DATA PROCESSING
Innovation Plaza
275 Paterson Avenue
Little Falls, NJ 07424

and are available for license exclusively from INNOVATION DATA PROCESSING.

FATS, FATAR and FATSCOPY are trademarks of INNOVATION DATA PROCESSING and are registered with the US Patent Office.

USA CONTACTS

Voice: 973-890-7300 Fax: 973-890-7147
E-mail: Sales: sales@fdrinnovation.com
Technical Support: support@fdrinnovation.com
Home page: http://www.innovationdp.fdr.com/index.cfm

EUROPEAN OFFICES:

France

Innovation Data Processing S.A.R.L.
191 Avenue Aristide Briand
94230 CACHAN
Tel: 01-49-69-94-02 Fax: 01-49-69-90-98
E-mail: frsupport@fdrinnovation.com
frsales@fdrinnovation.com

Netherlands

Innovation Data Processing
Brouwerstraat 8
1315 BP Almere
Tel: 036-534 1660 Fax: 036-533 7308
E-mail: nlsupport@fdrinnovation.com
nlsales@fdrinnovation.com

Germany

Innovation Data Processing
International Ltd.
Orleansstrasse 4a
D-81669 München
Tel: 089-489 0210 Fax: 089-489 1355
E-mail: desupport@fdrinnovation.com
desales@fdrinnovation.com

United Kingdom

Innovation Data Processing Ltd.
Clarendon House
125 Shenley Road
Borehamwood, Herts, WD6 1AG
Tel: 0208-905 1266 Fax: 0208-905 1428
E-mail: uksupport@fdrinnovation.com
uksales@fdrinnovation.com

© Copyright 1974, 2013
© INNOVATION DATA PROCESSING

Reproduction of this User Manual is prohibited except for licensed users for their internal use. A copy of this manual is provided on CD-ROM in Adobe PDF format for local printing, and is available at the Innovation FTP web site: http://www.innovationdp.fdr.com/ftp/ftp.cfm

Revised December 2013 V4.9L28

TABLE OF CONTENTS **Page No.**

1.0 INTRODUCTION

1.1 Summary of Modifications for Version 4.9 Level 28 1-1
1.2 Summary of Modifications for Version 4.9 Level 27 1-2
1.3 Summary of Modifications for Version 4.9 Level 26 1-3
1.4 Summary of Modifications for Version 4.9 Level 25 1-4

2.0 FATS, FATAR, and FATSCOPY Product Overview

2.0 FATS, FATAR, and FATSCOPY Product Overview 2-1

10.0 FATS FUNCTIONAL DESCRIPTION

10.0 FATS Functional Description 10-1

11.0 FATS TECHNICAL SUMMARY

11.0 FATS Technical Summary 11-1
11.1 General 11-1
11.2 Tape Labeling 11-4
11.3 Erasing Tapes 11-6
11.4 Tape Management Systems 11-8
11.5 Report Formats 11-9
11.6 Security 11-11

13.0 FATS EXECUTION JCL

13.0 FATS Execution JCL 13-1
13.1 Operator Control 13-4

14.0 FATS CONTROL STATEMENTS

14.0 FATS Control Statements 14-1
14.1 General 14-1
14.2 Operation Keywords 14-2
14.3 Operands 14-3
14.3.1 Retry/Error Operands 14-3
14.3.2 Volume Mounting Operands 14-5
14.3.3 Labeling Operands 14-7
14.3.4 Print Operands 14-8
14.3.5 Control Operands 14-9
14.3.6 Miscellaneous Operands 14-10
14.3.7 ANALYZE Operands 14-12

16.0 FATS EXAMPLES OF USAGE

16.0 Examples Of FATS Usage 16-1
16.1 Examples 16-3

TABLE OF CONTENTS Page No.

20.0 FATAR FUNCTIONAL DESCRIPTION

20.0 FATAR FUNCTIONAL DESCRIPTION 20-1

21.0 FATAR TECHNICAL SUMMARY

21.0 FATAR TECHNICAL SUMMARY 21-1
21.1 GENERAL 21-1
21.2 INPUT TAPE PROCESSING 21-2
21.3 MODIFICATION AND SCANNING 21-7
21.4 OUTPUT TAPE PROCESSING 21-8
21.5 TAPE MANAGEMENT SYSTEM SUPPORT 21-12
21.6 SECURITY 21-13

22.0 FATAR EXECUTION JCL

22.0 FATAR EXECUTION JCL 22-1

23.0 FATAR CONTROL STATEMENTS

23.0 FATAR CONTROL STATEMENTS 23-1
23.1 GENERAL 23-1
23.2 ANALYZE STATEMENT 23-2
23.2.1 INPUT TAPE OPERANDS 23-4
23.2.2 ERROR RECOVERY OPERANDS 23-7
23.2.3 OUTPUT TAPE OPERANDS 23-8
23.2.4 PRINT CONTROL OPERANDS 23-11
23.2.5 MISCELLANEOUS OPERANDS 23-13
23.3 MODIFICATION/SCAN STATEMENTS 23-15
23.3.1 PRINT STATEMENT 23-16
23.3.2 DROP STATEMENT 23-17
23.3.3 KEEP STATEMENT 23-18
23.3.4 REPLACE STATEMENT 23-19
23.3.5 SCAN STATEMENT 23-20
23.3.6 TABLE STATEMENT 23-21
23.3.7 RENAME STATEMENT 23-22
23.4 MODIFICATION/SCAN OPERANDS 23-24

24.0 FATAR SPECIAL CONSIDERATIONS

24.0 FATAR SPECIAL CONSIDERATIONS 24-1
24.1 VARIABLE SPANNED RECORDS 24-1
24.2 RECOVERING DATA FROM OVERWRITTEN TAPES 24-3
24.3 BENEFITS OF USING FATS/FATAR WITH A VTS 24-6

25.0 FATAR REPORT EXAMPLES

25.0 FATAR REPORT EXAMPLES 25-1

26.0 FATAR EXAMPLES OF USAGE

26.0 FATAR EXAMPLES OF USAGE 26-1
26.1 INDEX TO EXAMPLES 26-2
26.2 EXAMPLES 26-3

30.0 FATSCOPY FUNCTIONAL DESCRIPTION

30.0 FATSCOPY FUNCTIONAL DESCRIPTION 30-1
30.1 FATSCOPY SECURITY 30-3

31.0 FATSCOPY EXECUTION JCL

31.0 FATSCOPY EXECUTION JCL 31-1

32.0 FATSCOPY CONTROL STATEMENTS

32.0 FATSCOPY CONTROL STATEMENTS 32-1
32.1 GENERAL 32-1
32.2 COPY/SIM STATEMENT 32-2
32.3 SELECT/EXCLUDE STATEMENTS 32-16
32.4 RENAME STATEMENT 32-25
32.5 DATA SET NAME FILTERS (MASKS) 32-26
32.6 FATSCOPY SPECIAL CONSIDERATIONS 32-30

33.0 FATSCOPY EXAMPLES

33.0 FATSCOPY EXAMPLES 33-1

34.0 FATSCOPY PANELS

34.0 FATSCOPY PANELS 34-1

35.0 FATAUDIT FUNCTIONAL DESCRIPTION

35.0 FATAUDIT FUNCTIONAL DESCRIPTION 35-1
35.1 FATAUDIT EXECUTION JCL 35-2

40.0 FATS/FATAR/FATSCOPY MESSAGES AND CODES

40.0 MESSAGES AND CODES 40-1
40.1 RETURN CODES 40-1
40.2 ABEND CODES 40-2
40.3 CONSOLE MESSAGES 40-4
40.4 PRINT MESSAGES 40-8
40.5 FATS ACTION MESSAGES 40-46

TABLE OF CONTENTS Page No.

90.0 FATS/FATAR/FATSCOPY INSTALLATION

90.0 FATS/FATAR/FATSCOPY INSTALLATION 90-1
90.1 LOADING THE FATS/FATAR/FATSCOPY LIBRARIES 90-2
90.1.1 ELECTRONIC INSTALLATION 90-2
90.1.2 INSTALLATION FROM CD 90-8
90.2 ALTERNATE MANUAL INSTALLATION 90-11
90.3 ACTIVATING THE TAPE MANAGEMENT INTERFACE 90-12
90.4 CUSTOMIZING FATS, FATAR, AND FATSCOPY 90-14
90.4.1 DISPLAYING AND MODIFYING FATSCOPY DEFAULTS 90-14
90.4.2 SECURITY 90-17
90.4.3 OPTIONAL MODULES AND TABLES 90-18
90.5 INSTALLING THE ISPF DIALOG 90-19

93.0 FATS/FATAR ISPF INTERFACE

93.0 ISPF INTERFACE INTRODUCTION 93-1
93.1 FATS ISPF INTERFACE 93-8
93.2 FATAR ISPF INTERFACE 93-13
93.3 FATAR EXTENDED FUNCTIONS ISPF INTERFACE 93-20

THIS PAGE INTENTIONALLY LEFT BLANK.

1.1 SUMMARY OF MODIFICATIONS FOR VERSION 4.9 LEVEL 28

FATSCOPY IMAGE COPIES WITH TLMS

FATSCOPY can now be used to make image copies when the tape management system is TLMS, in addition to the previously-supported RMM and CA 1 image copies. See pages 32-8 and 32-31 for further details on using IMAGE.

USING SYSTEM LOGGER FOR AUDIT RECORDS

FATSCOPY can now write audit records to a system logger log stream. Concurrent jobs can write data to the same log stream. (Writing audit records to a sequential data set, which is still supported, requires a different audit data set for each concurrent job.) FATAUDIT can now read either sequential data sets or a log stream to produce audit reports; it can also read a log stream and write out the data to a sequential data set for long-term storage. See page 32-35 for information on using a system logger with FATSCOPY.

Using the system logger for audit data allows a customer to satisfy auditing requirements for recording copy activity to a single location, even when multiple copy jobs are running at the same time. The single log stream can be used to generate reports to satisfy auditing requirements.

FDREPORT NOW AVAILABLE TO FATSCOPY CUSTOMERS

FDREPORT, a companion product from Innovation Data Processing, is now available for use by FATSCOPY customers. FDREPORT can be used with RMM or CA 1 to query your tape management system using a broad range of powerful selection parameters, generating reports on your tape environment that you can use to determine which volumes you need to copy (selecting data sets expiring after a certain date, data sets created by certain programs or within a specified date range, etc.). Examples of using FDREPORT are on page 32-

For an EMC DLm, FDREPORT can also be used to generate scratch volume reports that can be used as input to the DLm's utility for scratching the backing disk space for logical tape volumes that are in scratch status. This is described on page 32-37.

If you have a FATSCOPY license, FDREPORT is available at no charge for a 90-day period. More information about FDREPORT is available at http://www.fdr.com/products/fdreport/.

REMINDER: INSTALLATION PROCEDURE FOR THE TAPE MANAGEMENT INTERFACE

The procedure for activating the tape management interface was changed in release 4.9.26. If you are upgrading from release 4.9.25 or earlier, you need to note the revised tape management interface procedure. Instead of running an assembly job from the ICL, you now run the new FATZAPOP program to define which tape management system you are using. This procedure is described in Section 90.3 on page 90-12. Alternatively, you may install it using the Option "I" on the Main Menu of the ISPF panels; this is described in Section 90.5.

1.2 SUMMARY OF MODIFICATIONS FOR VERSION 4.9 LEVEL 27

IMAGE COPIES WITH FATSCOPY

A new option, IMAGE, has been provided that directs FATSCOPY to make an "image" copy (an exact bit-for-bit copy) of a tape volume, similar to the image copy function of FATAR. The output tape contains an exact copy of the input tape, including the volume label, header and trailer labels, and all the data files. This can be used when you want to migrate tape volumes to a new output device, maintaining the original volsers and the position of each file on the volumes. FATSCOPY will update the tape management information for the volume to contain the output device information (such as device location and type) for the copied volume.

An image copy can be done only when SELECT ALLDSN is used, and when the tape management system is RMM or CA 1.

FATSCOPY can be used to copy OAM data sets when using IMAGE. (Otherwise, OAM data sets will automatically be excluded from selection by FATSCOPY.) This image copy must be done to a device with the same Media Type as that of the input tapes. After a volume is copied, an OAM UPDATE VOLUME operator command should be done to mark the volume as not writable .

See pages 32-8 and 32-31 for further details on using IMAGE.

FATSCOPY VOLUME RANGE: NUMVOLS

A new selection keyword, NUMVOLS, has been added to make it easier to specify a range of volumes to be selected by a FATSCOPY batch job when SELECT ALLDSN is used. See page 32-24 for details.

CATALOGING DATASETS WITH FATAR IMAGE COPY

Cataloging will now be done by a FATAR image copy when CAT=RECAT is used. This allows the z/OS catalog to be updated to contain the correct device type for the copied data sets.

NEW PROCEDURE FOR ACTIVATING THE TAPE MANAGEMENT INTERFACE

The procedure for activating the tape management interface was changed in release 4.9.26. If you are upgrading from release 4.9.25 or earlier, you need to note the revised tape management interface procedure. Instead of running an assembly job from the ICL, you now run the new FATZAPOP program to define which tape management system you are using. This procedure is described in Section 90.3 on page 90-12. Alternatively, you may install it using the Option "I" on the Main Menu of the ISPF panels; this is described in Section 90.5.

1.3 SUMMARY OF MODIFICATIONS FOR VERSION 4.9 LEVEL 26

FATAR/FATSCOPY Changes

NEW PROCEDURE FOR ACTIVATING THE TAPE MANAGEMENT INTERFACE

The procedure for activating the tape management interface has changed. Instead of running an assembly job from the ICL, you now run the new FATZAPOP program to define which tape management system you are using. This procedure is described in Section 90.3. Alternatively, you may install it using the Option "I" on the Main Menu of the ISPF panels; this is described in Section 90.5.

FATSCOPY Changes

SIMPLIFIED COPYING OF ENTIRE VTS PHYSICAL VOLUMES

The new ALLPHYS= keyword can be used to specify a *logical* volume and have FATSCOPY select *all* the logical volumes on the same *physical* tape on a supported tape-based virtual tape system (VTS). This allows you to efficiently copy physical tapes when you do not know the physical volser containing one of the logical volumes you want to copy.

RESTART AFTER ABEND OR OPERATOR CANCEL

When an operator issues a CANCEL command for a FATSCOPY job that uses a DSNTABLE DD statement, or when certain ABENDS occur, FATSCOPY will now save a list of the data sets that were selected but not yet copied in the DSNTABLE data set. This data set can be used in a subsequent RESTART job to continue copying the data sets. (Previously, an operator STOP command was needed to interrupt a FATSCOPY job and create the DSNTABLE data set.)

NEW OPTION TABLE

You can now configure the default values for many FATSCOPY parameters on your system using the FATZAPOP program, described in Section 90.4.1. The default values can be displayed in the FATSCOPY output listing (using the new ECHO keyword) or with FATZAPOP (using the PRINT keyword). An individual job can still override any default by specifying a corresponding FATSCOPY keyword.

NEW ISPF PANEL TO SET OPTIONS

A new ISPF panel has been provided to set FATSCOPY installation defaults for the tape management interface and for JCL defaults.

RETRIES WHEN RMM HOUSEKEEPING IS RUNNING

When the default RMMFAIL=CONTINUE is used, FATSCOPY will proceed to the next data set without attempting to retry any RMM updates. This is the way that RMMFAIL worked in version 4.9.20.

CAT=ONLY NOW THE DEFAULT FOR SELECT CATDSN

The default value for the CAT= parameter when SELECT CATDSN is used has been changed. If you do not explicitly specify a CAT= parameter, the delivered default value is now CAT=ONLY, matching the default used when SELECT ALLDSN is used.

1.4 SUMMARY OF MODIFICATIONS FOR VERSION 4.9 LEVEL 25

FATSCOPY Changes

IMPROVED VIRTUAL TAPE SERVER PERFORMANCE

When copying tape volumes from a tape-based virtual tape system (VTS), considerable time can be saved by minimizing the number of physical tape mounts needed. With real (cartridge) tapes, FATSCOPY has always sorted a list of selected data sets to minimize tape mounts. With virtual tape systems, many logical volumes may be located on a single high-capacity physical volume. Additional time savings can be achieved by obtaining information from the VTS about which logical volumes are located on each physical volume, to avoid tape mount thrashing.

FATSCOPY Version 4.9 Level 25 has the capability of obtaining mapping information from an IBM (TS7700 or 3494) or Oracle StorageTek VSM virtual tape system. With the new keywords VIRTTYPE= and PHYSVOL=, the user can specify a list of physical volumes, and FATSCOPY will select all the virtual volumes on those physical volumes and copy them, ensuring only 1 physical tape mount needs to be done for the set of virtual volumes on each physical volume. This would be most useful when you want to migrate from one system to another, completely copying all the files from each physical volume on the source VTS.

RETRIES WHEN RMM HOUSEKEEPING IS RUNNING

When the default RMMFAIL=CONTINUE is used, you can now specify that FATSCOPY should retry an RMM update that has failed due to RMM housekeeping running at the same time as FATSCOPY. The number of retries is specified by the RETRY= parameter. (Previously, with RMMFAIL=CONTINUE FATSCOPY would proceed to the next data set without attempting any RMM update retries.)

SUPPRESS COPYING USER LABELS

The new keyword SUPULAB will prevent FATSCOPY from copying user labels from the input data set to the output data set. This can be useful when another product writes its own user labels on the output tape, such as for storing encryption keys.

GROUPING DATA SETS BY EXPIRATION DATE OR INPUT VOLSER

The new keyword GROUPBY= can be used to change the way that FATSCOPY groups the output data sets. You can specify that the output data sets should be grouped by the expiration dates of the input files, or by the current grouping of the files on the input volumes.

FATS Changes

INCREASED BLOCKSIZE FOR FATS CERTIFY

When your operating system supports large blocks, FATS will now write blocks as large as 262,144 (256K) bytes during certification. This can reduce the time needed to certify a tape by up to 25 percent. (The previous blocksize limit was 65,535 bytes.)

2.0 FATS, FATAR, AND FATSCOPY PRODUCT OVERVIEW

FATS (sections 10 through 16) is used to perform several basic functions with one or more tapes:

- Certify the usability of cartridge tapes
- Label tapes
- Erase tapes
- Invoke the companion product FATAR (if licensed) to perform detailed data analysis and copy one or more tapes

FATAR (sections 20 through 26) is a multi-purpose tape utility to:

- Examine the labels and data on a tape (ANALYZE function)
- Verify that specified fields on a tape contain valid data
- Copy one tape to another. Copies can:
  - Be an exact bit-for-bit copy, including the volume serial number (IMAGE copy)
  - Use a different label type (such as AL-to-SL copy)
  - Use a different block size on the output tape (REBLOCK function)
  - Use different names for the output data sets (RENAME function)
- Copy a tape with I/O errors to another tape, applying user-specified modifications to alter or drop data blocks which caused those errors

A product license for FATAR includes a license for FATS.

FATSCOPY (sections 30 through 35) is a full-featured tape copy utility to:

- Automate the stacking of data sets by copying files from several underutilized or smaller volumes onto a smaller number of output tapes
- Migrate files from one device type to another
- Select (by volser or data set name mask) data sets to be copied, and apply additional filtering to limit the data sets selected by criteria such as job name, creation date, expiration date, etc. (unlike FATAR, where you must always directly specify which volsers to process)
- Make image (exact bit-for-bit) copies of volumes
- Use a different block size on the output tape (REBLOCK function)
- Use different names for the output data sets (RENAME function)
- Copy tapes from some proprietary products, such as ABR Archive backups and HSM ML backups, that maintain information about the location of data sets in external data bases
- Set the expiration dates of input data sets after they have been copied so they will be expired by your tape management system
- Update the tape management information for the output data sets, copying information from the tape management records for the input data sets

A product license for FATSCOPY includes a license for FATAR and FATS. In addition, a product license for FATSCOPY makes you eligible for a 90-day trial of FDREPORT, Innovation's powerful reporting utility that can give you insights into the contents of your tape environment.

The following additional documentation is available at the Innovation web site:

- **FATSCOPY How-To-Build-A-Job Guide** – a step-by-step tutorial for learning how to use FATSCOPY. This PDF document asks you a series of questions about your copy task, and by clicking on the links corresponding to your answers you are shown what statements should be added to a batch job.

- **FATSCOPY Quick Start Guide** – a simple description of how to get started with FATSCOPY using the ISPF panels.

- **FATSCOPY Concepts & Facilities Guide**

- **FATSCOPY Product Demo**

10.0 FATS FUNCTIONAL DESCRIPTION

WHAT IS FATS USED FOR?

FATS has several specific applications, including:

- Certifying the usability of physical tapes, both new and old, by writing a pattern on the full length of the tape; this is equivalent to the function of hardware tape certifiers. This helps to avoid write data errors at application run time by identifying potentially defective tapes.
- Verifying the readability of critical or archival data tapes, to avoid read data errors at application run time. Tapes with read errors may be recovered using FATS' companion product FATAR.
- Labeling tapes with specified volume serials. This can be done by itself, or in combination with a certification process.
- Erasing data from a tape, to insure that confidential information is not left on a tape (as when a tape volume is being sold or shipped to another location). This is similar to stand-alone tape degaussers, but is superior since tape serials can be verified and preserved. FATS can erase only the residual data on a tape through the use of the RESIDUAL keyword. See Section 14.3.6 for a description of this keyword.
- Interfacing with its companion product FATAR to perform detailed data analysis and copying of tapes.

Regular use of the certification and verification functions of FATS can significantly reduce the number of application abends due to tape data checks.

TAPE OPERATIONS

Depending on the command statements provided, FATS will perform any of its supported functions on up to 9 tape drives simultaneously. The same operation could be performed on all of the allocated tape drives, or, if desired, different functions could be done on different drives. FATS uses an internal task control so that all of these operations can proceed independently of one another. By the use of FATS parameters, FATS can be instructed to request multiple tapes on a drive, calling either for scratch tapes or specific volume serials, so that the same operation can be performed on many tapes using one tape drive. FATS will terminate when instructed by the operator or when all requested functions are completed.

SUPPORTED DEVICES

FATS will function on any IBM or IBM-compatible tape drive or virtual tape system (VTS) supported by the host z/OS system.

PROCESSING SPEED

The speed at which FATS can process a tape is dependent on three variables: the speed of the tape drive, the contention for the tape channel and control unit, and the quality of the tapes being tested. When testing tapes suspected of being exceptionally poor, a low permanent error retry level can be specified to accelerate processing. FATS may also contend with itself if multiple tapes being tested concurrently are on the same tape control unit or channel; when processing more than three tapes in one FATS job (except for ERASE functions), it is wise to process multiple tapes using more than one channel and control unit, if available.
When certifying or erasing a single tape, you can expect elapsed times similar to:

Certify a 3490E Enhanced (1100 ft) cartridge	10.0 minutes
Erase a 3490E Enhanced cartridge	7.2 minutes
Certify a 3590 (Magstar) 10GB cartridge	26.1 minutes
Erase a 3590 (Magstar) cartridge	19.6 minutes

Erase times will be independent of other tape activity, but certify times will increase if there is other tape activity (including other FATS or FATAR activities) on the same tape channel or control unit. Read times will vary depending on the amount of data and blocksize on the tape and cannot be estimated.

11.0 FATS TECHNICAL SUMMARY

11.1 GENERAL

THE FATS PROGRAM

FATS (Fast Analysis of Tape Surfaces) is a utility program for certifying, verifying, labeling, and erasing magnetic tapes.

It is recommended that FATS be run using REGION=0M.

FATS must be link edited and executed as an authorized program.

CONTROL AND REPORT I/O

FATS accepts its control statement input from DD statement SYSIN. The control statements are 80-byte records, of which only columns 1 to 71 may contain information.

Report output is directed to several print data sets. DD statement SYSPRINT is required, and will contain general messages and control statement listings. Up to 9 optional SYSPRINx DD statements may be included (where "x" is a digit from 1 to 9 and corresponds to the 9 tape drives used by FATS); if present, detail reports on the results from the associated tape drive will be printed on SYSPRINx, but if omitted, the detail reports will go to SYSPRINT. DD statement SUMMPRT is also optional; if present, it will receive a summary report on each tape volume processed by FATS; if absent, the summary report will go to the associated SYSPRINx or to SYSPRINT.

TAPE OPEN PROCESSING

FATS always opens every tape with a standard data management OPEN, so that label processing will be performed (unless bypassed by JCL or control statement options), and tape management and security systems, if present, will be invoked. So, for all FATS operations, you have assurance that the tapes mounted are the proper tapes and that they may be written on or read from as long as you do not bypass these checks yourself.

If label processing is bypassed (LABEL=(,BLP) in JCL, or the BLP control statement option), the FATS option VALIDATE= may be used to check the volume serial and expiration date on tapes used by FATS.

FATS CONTROLS

The system operator or tape librarian can exercise control over FATS by optionally replying to an outstanding WTOR (write-to-operator-with-reply) or by issuing a console STOP or MODIFY command. Use of the WTOR=YES operand (the default) will cause a WTOR to be issued to the tape pool console (route code 3) when FATS is executing. The operator can terminate FATS at any time by replying to this message; FATS can also be instructed to abandon processing of a particular tape (if, for example, the tape is too damaged to mount). Alternatively, by using the MODIFY=YES parameter (default is MODIFY=NO), the operator or librarian may use console STOP and MODIFY commands to gain the same control over FATS.

LABELING TAPES

FATS can label new tapes or relabel existing tapes during the certification process. Labels can be IBM standard or ANSI V3 labels. FATS can also create unlabeled tapes. See Section 11.4 for more information.

WORM AND VIRTUAL TAPES

FATS will reject an attempt to certify a WORM (Write Once Read Many) cartridge, since this would make the cartridge unusable for customer data.

ISPF PANEL SUPPORT

Extensive ISPF Panels are available to perform many of the FATS/FATAR/FATSCOPY functions such as labeling, certifying and copying tapes. See Section 93.1 a description of the FATS ISPF Interface.

ISPF PANELS FOR CERTIFICATION

Below are examples of the FATS ISPF panels for certification of new and existing tapes. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

FATS CERTIFY NEW TAPES (OPTION 1)

```
  ------------------------ FATS CERTIFY NEW TAPES -----------------------------
COMMAND ===>                                                Enter Q for   TMS Query

TAPE1  DD Dsname=  BAB.FATSOUT.JOBD                                 <
          Unit  =( TAPE     )
          Label =( ,BLP,EXPDT=98000                     )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN  - Maximum number of volumes : ___
NOLABEL  - Create no label (NL) tapes: NO    (no/yes)
ANSI     - Create ANSI labeled tapes : NO    (no/yes)
VALIDATE - Verify Volser/Expired Date: NO    (no/yes)

Display more FATS Operands?: NO  (yes/no)
```

FATS CERTIFY EXISTING SCRATCH TAPES (OPTION 2)

```
------------------------ FATS CERTIFY EXISTING TAPES ------------------------
COMMAND ===>                                                Enter Q for   TMS Query


TAPE1  DD Dsname=  BAB.FATSOUT.JOBD                                 <
          Unit  =( TAPE     )
          Label =( ,SL                                  )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN - Maximum number of volumes : ___
NOLABEL - Create no label (NL) tapes: NO  (no/yes)
ANSI    - Create ANSI labeled tapes : NO  (no/yes)
VALIDATE- Verify Volser/Expired Date: NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)
```

ISPF PANEL FOR VERIFICATION

Below is an example of the FATS ISPF panel for verification of active tapes. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

FATS VERIFY (READ) OF EXISTING DATA ON TAPE (OPTION 5)

```
  ---------------------- FATS VERIFY (READ) TAPES --------------------------
COMMAND ===>                                                Enter Q for   TMS Query

TAPE1  DD Dsname=  FATS                                     <
          Unit  =( TAPE     )
          Label =( ,BLP,EXPDT=98000                    )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN - Maximum number of volumes          : ___
FILES   - Number of Physical files to be read: 3
VALIDATE- Verify Volser/Expired Date         : NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)
```

11.2 TAPE LABELING

FATS labels a tape by writing an IBM standard (SL) or ANSI V3 (AL) label set.

FATS can write labels during certification or erase of a tape, and can also label tapes as a separate function (at considerable savings in elapsed time). Because FATS can request mounting of a large number of volumes, it is ideal for initializing a sequence of new tapes, or re-labeling old tapes with new serials.

The serial number can be specified in FATS control statements, in JCL, or via the operator's console. However, for ease in certifying tapes that are already labeled, the FATS default option SAVLAB will read the existing volume serial from the tape being certified and re-label the tape with it, thus "saving" the volume serial.

FATS can also create unlabeled tapes, writing only a tape mark at the beginning of the tape.

When labeling new tapes, FATS can be instructed to mount a range of sequentially numbered tapes with simple control statements.

FATS LABEL SET

When labeling tapes, either directly via the LABEL function, or as an option of the WRITE or ERASE functions, FATS will write the following records:

VOL1 label - with the specified volume serial, and optional OWNERID
HDR1 label *
HDR2 label - showing RECFM=U and BLKSIZE=32760
Tape Mark
Tape Mark - empty data file
EOF1 label *
EOF2 label - showing RECFM=U and BLKSIZE=32760
Tape Mark
Tape Mark - end of tape

* The HDR1 and EOF1 labels will contain the data set name that appeared in the TAPEx DD statement, except that if LABEL=EXPDT=98000 also appeared on that DD statement, the dsname will consist of 17 "0"s (zero character) so that the tape will be immediately acceptable as a scratch tape for tape management systems. The creation and expiration dates are set to the current date.

The labels will be IBM standard labels, except when the ANSI operand is specified, in which case they will be ANSI V3 standard labels.

NOTE: The labels written by FATAR differ from IEHINITT and some other tape labeling products, which only write a VOL1, HDR1, and one Tape Mark. Although this minimal label set is valid, it may cause some tape mapping software to attempt to read past the tape mark and get a read error.

WORM TAPES

The IBM 3592 and TS1120 and Sun tape drives support WORM (Write Once Read Many) cartridges. The drive allows data to be added to a WORM tape but existing data can only be read, never updated. However, the drive does allow header and trailer labels to be updated when appropriate.

FATS can be used to label a WORM tape, or to relabel a WORM tape that has never been used for data. Because the way that the drive supports WORM, FATS will detect that a cartridge is a WORM and will write labels in a form consistent with that support (only a VOL1, HDR1, and a single tape mark). This allows the cartridge to be opened for output to create the first file.

ISPF PANEL FOR LABELING

Below is an example of the FATS ISPF panel for labeling of new tapes or relabeling old tapes with new serial numbers. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

FATS LABELING OF NEW TAPES (OPTION 3)

```
------------------------- FATS LABEL NEW TAPES -----------------------------
COMMAND ===>                                              Enter Q for   TMS Query

TAPE1  DD Dsname=  FATS                                        <
          Unit  =( TAPE     )
          Label =( ,BLP,EXPDT=98000                      )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN - Maximum number of volumes : ___
NOLABEL - Create no label (NL) tapes: NO  (no/yes)
ANSI    - Create ANSI labeled tapes : NO  (no/yes)
VALIDATE- Verify Volser/Expired date: NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)
```

11.3 ERASING TAPES

ERASING TAPES

All physical tape drives support a "Data Security Erase" feature, that can be used to erase data on a tape at high-speed without tying up the tape control unit. This Erase function can therefore clear all data from a tape without impacting other jobs using tape drives on the same tape control unit. The erase takes place at maximum tape write speed (which varies by tape drive model).

FATS fully supports the Erase function. When invoked, FATS ERASE will write labels on the tape (if indicated), and then start the hardware erase function that will continue without intervention from the CPU or control unit until the end-of-tape is reached. For tape cartridge systems, ERASE involves writing a random pattern on the tape for complete security. For round tapes, the tape drive's erase head is used to erase the tape to "unrecorded" status.

By default, ERASE will preserve the existing volume serial on the tape, performing a FATS label function before starting the hardware erase (see Section 11.4). Optionally, you can relabel it with a new serial number, or create an unlabeled tape.

By default, ERASE will erase the entire tape. However, the RESIDUAL operand causes FATS to locate the end of the last existing file on the tape and erase the remainder of the tape. This allows you to securely erase any residual data from a non-scratch tape. This can be useful when a tape will be sent off-site, to prevent anyone from accessing sensitive data left over from a previous use of the tape.

This ERASE function may be important to installations that are upgrading to new tape technology and are planning to sell or discard many of their old tapes. **ERASE may be used to insure that all company confidential data is gone from those tapes before they leave the site.** Many users first run a FATS CERTIFY before a FATS ERASE to ensure that none of the data will be readable.

Note that the ERASE function, because it is a single very long I/O operation, may receive the IBM message

IOS071I uuuu DEVICE END MISSING

This is normal and can be ignored. You may also receive the IBM message

IOS000I devn,chpid,SIM

The IOS000I message may be followed by a system 714-0C abend.

You can eliminate these messages and the system 714 abend by increasing the MIH (Missing Interrupt Handler) timeout value for the tape drives (this may be modified in the IECIOSxx member of PARMLIB or by a SETIOS console command). Tape cartridge drives will display the message "ERASE" or "ERASING" while the ERASE operation is in progress.

WARNING: This function will erase all data on a tape/cartridge with no possibility of recovery. Use caution when using ERASE.

WORM AND VIRTUAL TAPES

FATS will reject an attempt to erase a WORM (Write Once Read Many) cartridge, since this would make the cartridge unusable for customer data.

FATS should not be used to erase virtual tapes (known as VTS or VSM). Some virtual tape devices may reject the ERASE; even if it works, it will be meaningless since there is no real tape surface to erase.

FATS can be used to erase the physical tapes from a Virtual Tape Server, but the volumes need to be ejected from the VTS before they can be erased. If there are tape drives available outside the VTS, the ejected volumes can be erased using the standalone drive. If no tape drives are

available outside the library, the ejected volumes need to be reentered into the tape library as scratch volumes. You can then erase the physical tapes.

ERASING ENCRYPTED TAPES

Using FATS to erase data from encrypted tapes requires different JCL than that used for unencrypted tapes because encrypted tape drives rewrite the header labels for the tape as soon as a tape is opened for output. If ERASE is specified and LABEL=(x,BLP) or LABEL=(x,NL) is specified, FATS will abend with a U0888. If ERASE RESIDUAL and LABEL=(,SL), LABEL=(1,SL), LABEL=(x,BLP), or LABEL=(x,NL) is specified on the TAPEx DD for a tape that is encrypted, FATS will issue a U0888 abend to prevent all the data from being wiped out when the tape is opened. If you want to erase all the data from a tape that contains encrypted data, specify LABEL=(,SL) on the TAPEx DD.

If you want to erase the residual data from a tape that currently contains encrypted data, you should consider the following:

1) If all the residual data on a tape is encrypted, there's no need to erase the data as the information about the key used to encrypt the residual data is lost when the tape was written over.

2) If you need to erase the residual data on a tape that previously contained unencrypted data, but now contains encrypted files you want to preserve, you need to code LABEL=(x,SL) on the TAPEx DD. For 'x', substitute a value 1 higher than the number of files that you want to preserve. For example, if a tape has 2 encrypted files and you want to erase the remainder of the tapes, specify LABEL=(3,SL) on the TAPEx DD. If you specify LABEL=(,SL), FATS will abend with a U0888. After the tape is erased, it will contain the files you wanted to preserve, followed by a dummy file containing only header labels.

ISPF PANEL FOR ERASE

Below is an example of the FATS ISPF panel for erasing tapes. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

FATS ERASE FUNCTION (OPTION 4)

```
---------------------- FATS ERASE OLD TAPES -------------------------------
COMMAND ===>                                          Enter Q for   TMS Query

TAPE1  DD Dsname=  FATS                                      <
          Unit  =( TAPE     )
          Label =( ,SL                                  )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN - Maximum number of volumes     : ___
VALIDATE- Verify input Volume Serial    : NO  (yes/no)
SAVLAB  - Preserve volume serial?       : YES (yes/no)
NOLABEL - Create no label (NL) tapes    : NO  (yes/no)
ANSI    - Create ANSI labeled tapes     : NO  (yes/no)
RESIDUAL- erase residual data after EOD: NO  (yes/no)

Display more FATS Operands?: NO  (yes/no)
```

11.4 TAPE MANAGEMENT SYSTEMS

TAPE MANAGEMENT SYSTEMS

Although FATS has no formal interface to any tape management system, it is designed to be compatible with them. FATS takes the following actions for the benefit of tape management systems:

- Tapes will always be opened with the label type indicated in the JCL (unless the FATS BLP operand is used). If tapes are opened as labeled, the tape management system knows the volume serial of the tape and can approve or disapprove its use. If they are opened as unlabeled, an operator response may be required to provide the volume serial for the tape management system.
- When tapes are opened for output (certification, labeling or erasure), the expiration date of the data set will be set to the current date, so that the data set will be immediately available as a scratch tape.
- If labels are written on the tape, the data set name will be the name given in the JCL for the tape, unless EXPDT=98000 is given when a data set name of 17 "0"s is used. Some tape management systems will not allow a tape to be used as scratch if the name in the tape labels is different from the name recorded in its database, unless the name is 17 "0"s indicating it has been initialized.

Most tape management systems provide the ability to bypass their operation on a particular TAPE DD Statement. In many cases, EXPDT=98000 in the JCL is used for this bypass; check your tape management documentation to be sure. This may be required for certain operations with FATS. For example, if multiple volumes are being verified (read) by FATS, and their data set names differ, FATS will open them all with the data set name on the TAPE DD Statement; this may fail unless the tape management system is bypassed.

Note: the companion products FATAR and FATSCOPY have tape management system interfaces with IBM's RMM and CA Technologies' CA 1 and TLMS.

11.5 REPORT FORMATS

DETAIL REPORT

The FATS detail report contains one line for each significant event that occurs on a tape being processed by FATS. Significant events include labels written or saved, permanent or temporary data checks, end-of-tape (TAPE INDICATE), tape marks read, and certain error conditions. Each detail line will include:

- The TAPE DD name (ID), e.g., TAPE1
- The tape unit address (UCB) as a 4-digit hex number, e.g., 0381 or 4751
- The tape volume serial (LABEL)
- The FATS operation keyword (OPTION), e.g., WRITE
- The PASS number (always 1).
- The file number (FILE NO) and record number (RECORDS) within the file (for READ only)
- The LOCATION, in feet or meters from the beginning of the tape, of the event
- The LENGTH, in inches or centimeters, of any permanent data check (if contiguous data checks occur, this length is cumulative)
- The number of RETRIES attempted for any data check
- An ACTION message detailing the type of event (these messages are listed in the "Messages and Codes" section of this manual).

From this detail report, it is possible to see where the errors are and how they are grouped. For a WRITE operation, the location listed for TAPE INDICATE (end-of-tape) shows the total length of the tape. For a READ operation, the line for TAPE MARK gives the number of records in the file and length of tape read to that point (the length is cumulative, so you must subtract to get individual file lengths).

By default, FATS will print a blank line for every four detail report lines, and will print a blank line whenever a new volume is mounted on a tape unit. If the detail report is printed on a separate report file (DD name SYSPRINx rather than SYSPRINT), options are available to control the number of detail lines grouped together, and to skip to a new page for a new volume.

SUMMARY REPORT

The FATS summary report will contain several lines for each tape volume processed by FATS. The first line will contain ID, UCB, LABEL, and option (same as in the detail report above), plus the completion code (NORMAL or ABNORM), and the total length, in feet or meters, of the tape as processed by FATS.

The second line will appear once and gives the number of data checks encountered on the tape summarized by the number of retries performed for each. Counters for retry levels 1 through 10 appear on this line; the word "*PERM" appears above the column that is considered a permanent error by FATS (the value of the RETRY= parameter); non-zero counts for any lesser retry values indicate temporary errors. If RETRY= had a value greater than 10, this line will contain only the total number of temporary and permanent data checks, with appropriate headings above.

FATS CONTROL REPORT

```
FAST ANALYSIS OF TAPE SURFACES - CONTROL - FATS  VER 4.9.28 INNOVATION DATA PROCESSING  AUTHORIZED  08/11/2013  PAGE   1
FATS100 FOR INSTRUCTIONS AND MESSAGES AND CODES USE PARM=I
FATS110   WRITE(1) MAXERR=3,
FATS110     VOL=(001210,001540,003320)

FATS111 CHARACTERISTICS OF THE TAPES TO BE ANALYZED
FATS111   ID    PASS  OPTION LABEL FILES RETRY MODE BPI     GAP   REWIND THRESHOLD MAXERR BLKSIZE
FATS111  TAPE1   1    WRITE   SAV         10    D4  038000  3040    YES    01000    00003  65528
FATS301 END OF REPORT
```

FATS DETAIL REPORT

```
FAST ANALYSIS OF TAPE SURFACES - DETAIL  - FATS  VER 4.9.28 INNOVATION DATA PROCESSING  AUTHORIZED  08/11/2013  PAGE   1
MESSAGE   ID    UCB  LABEL   OPTION PASS FILE NO  RECORDS   LOCATION  LENGTH  RETRIES    ACTION
FATS107  TAPE1 0381  001210  WRITE    1                        0 FT                     LABEL SAVED
FATS107  TAPE1 0381  001210  WRITE    1                        5 FT             05      TEMP DATA CHECK
FATS107  TAPE1 0381  001210  WRITE    1                       56 FT             05      TEMP DATA CHECK
FATS107  TAPE1 0381  001210  WRITE    1                       67 FT             08      TEMP DATA CHECK

FATS107  TAPE1 0381  001210  WRITE    1                      104 FT    1 IN     10      PERM DATA CHECK
FATS107  TAPE1 0381  001210  WRITE    1                      104 FT    2 IN     10      PERM DATA CHECK
FATS107  TAPE1 0381  001210  WRITE    1                      105 FT    3 IN     10      MAX ERROR EXCEEDED *

FATS107  TAPE1 0381  001540  WRITE    1                        0 FT                     LABEL SAVED
FATS107  TAPE1 0381  001540  WRITE    1                      384 FT             02      TEMP DATA CHECK
fFATS107  TAPE1 0381  001540  WRITE    1                      506 FT                      TAPE INDICATE *****

FATS107  TAPE1 0381  003320  WRITE    1                        0 FT                     LABEL SAVED
FATS107  TAPE1 0381  003320  WRITE    1                      510 FT                     TAPE INDICATE *****
FATS301 END OF REPORT
```

FATS SUMMARY REPORT

```
FAST ANALYSIS OF TAPE SURFACES - SUMMARY - FATS  VER 4.9.28 INNOVATION DATA PROCESSING  AUTHORIZED  08/11/2013  PAGE   1
MESSAGE  ID    UCB  LABEL  OPTION  COMP    LENGTH   ***************  DATA CHECKS BY NUMBER OF RETRIES  ***************
                                    CODE              ONE    TWO  THREE   FOUR   FIVE    SIX  SEVEN  EIGHT   NINE    TEN
FATS300 TAPE1 0381  001210  WRITE  ABNORM  00105 FT                                                                 *PERM
FATS300                                   PASS 1         0     0      0      0      2      0      0      1      0      3

FATS300 TAPE1 0381  001540  WRITE  NORMAL  00506 FT                                                                 *PERM
FATS300                                   PASS 1         0     1      0      0      0      0      0      0      0      0

FATS300 TAPE1 0381  003320  WRITE  NORMAL  00510 FT                                                                 *PERM
FATS300                                   PASS 1  NO DATA CHECKS ON THIS TAPE
FATS301 END OF REPORT
```

11.6 SECURITY

STANDARD FATS SECURITY

FATS will issue an OPEN for the input and output tapes, so your security system will be invoked. If you have configured it to protect tape data sets and/or tape volumes, those security rules will apply, so the TSO userid under which FATS is run must be authorized to read the input volumes and data sets and write to the output volumes and to create the output data set names used. However, see the notes on BLP below.

PROTECTING FATS FUNCTIONS

FATS allows you to use your security system to control which users are authorized to execute each of the FATS functions. For example, you might authorize all operators to LABEL tapes, but restrict the ability to ERASE a tape to supervisors. **These security checks are disabled by default; see Section 90 for instructions on enabling them.**

Use of the FATS functions is controlled by resources defined in the RACF FACILITY class (or its equivalent in other security systems). Consult your IBM RACF or other security vendor documentation for the procedure for defining such FACILITY class resources. Once defined, you must grant READ authority to the resource for users authorized to use it.

For compatibility with earlier releases of FATS, the default is to allow all users to execute all functions. If your security system indicates to FATS that a given resource name is not defined, all users will be able to use that function. If you define some but not all of the resources, only those functions are protected. For example, if you don't define FATS.READ, all users can execute FATS READ functions.

The five FACILITY class resources that may be checked by FATS are:

FATS.READ	use the READ (verify) function
FATS.WRITE	use the WRITE (certify) function
FATS.LABEL	use the LABEL (label tapes) function
FATS.ERASE	use the ERASE (erase tapes) function
FATS.BLP	use the BLP operand or LABEL=(,BLP) in JCL.

As each control statement is processed, FATS will issue a RACROUTE to verify that the user has at least READ authority to the proper resource name. The step will fail if RACROUTE returns a code of 8 or higher, meaning that the resource is defined to your security system but the user associated with the FATS step does not have authority to it.

There is no FACILITY resource for the ANALYZE function. ANALYZE invokes FATAR, which has its own set of security checks as described in Section 21.6.

BYPASS LABEL PROCESSING (BLP)

When label processing on input or output volumes is bypassed using the BLP option in JCL or the BLP operand on the FATS statements, volume labels and data set names are not verified by OPEN and so authorization to the volume and data sets cannot be guaranteed. If anything, your security system will check authority to the volser and dsname provided in the JCL when BLP is used; these may not match the actual tape. For this reason, most security systems allow your installation to restrict the use of BLP.

However, some FATS functions (labeling or certifying new tapes) **require** the use of BLP. Innovation suggests that you provide BLP authority to a limited number of users (such as operators, tape librarians, and/or system programmers) so that they can run these functions when required.

Resource FATS.BLP controls the ability to use the BLP operand on FATS control statements and the ability to put LABEL=(,BLP) on a TAPEx DD statement. When opening a tape, FATS will check if BLP is being used, from either source, and will fail the step if the resource is defined but the user is not authorized. If the user is authorized to FATS.BLP but you have a general control on the use of BLP defined in your security system, the user must also be authorized to that resource.

13.0 FATS EXECUTION JCL

To execute FATS, the following JCL statements are required:

EXEC STATEMENT

The EXEC statement specifies the FATS program name (PGM=FATS), memory requirements (REGION=, if your installation defaults are insufficient), and parameter field (PARM=). It is recommended that you specify REGION=0M.

If PARM=I is specified on the EXEC statement, FATS will print on SYSPRINT a brief summary of its control statements and messages.

STEPLIB/JOBLIB DD STATEMENT

A STEPLIB or JOBLIB DD Statement will be required if FATS has been linkedited into a private library. It can be omitted if FATS is in a system library that can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). The library must be an APF authorized library.

SYSPRINT DD STATEMENT

SYSPRINT receives the listing of all FATS control statements, a summary of the operations to be performed, and messages about major errors. In addition, it may receive the detail reports for each tape drive and the summary report if the DD Statements for those reports are absent. SYSPRINT is normally allocated to a SYSOUT data set. Its DCB attributes are RECFM=FBA, LRECL=121. If a blocksize is specified, it must be a multiple of 121; otherwise it will default to 121 for SYSOUT or 1210 for other devices.

SYSPRINx DD STATEMENT

SYSPRINx receives the detail report for the operations performed against tapes mounted on DD statement TAPEx, where "x" is a digit from 1 to 9. This allows the detail reports for simultaneous FATS operations to be printed separately. SYSPRINx is normally allocated to a SYSOUT data set. If SYSPRINx is not present for a particular TAPEx, its detail report will go to SYSPRINT; however, SYSPRINx DD statements are required when running ANALYZE operations (FATAR under FATS). DCB attributes for SYSPRINx are the same as those for SYSPRINT.

SUMMPRT DD STATEMENT

SUMMPRT receives the summary report, consisting of several lines summarizing the results of FATS operation against each tape processed. SUMMPRT is normally allocated to a SYSOUT data set. If SUMMPRT is not present, summary reports will go to the SYSPRINx data set associated with the TAPEx DD Statement on which the tape was mounted; if it is also absent, summary reports go to SYSPRINT. DCB attributes are the same as for SYSPRINT.

TAPESUMM DD STATEMENT

When running ANALYZE operations (FATAR under FATS), TAPESUMM may optionally be included to receive the FATAR tape summaries, including data set name, tape label information, block and byte counts, lengths, minimum/maximum/average blocksizes and error counts. DCB attributes are the same as for SYSPRINT.

SYSUDUMP DD STATEMENT

SYSUDUMP requests an abend dump if major errors occur. Most internal abends in FATS are for the user's information only and do not cause dumps. SYSUDUMP is usually allocated to SYSOUT. If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced:

```
//ABNLIGNR  DD  DUMMY
```

TAPEx DD STATEMENT

TAPEx defines a tape unit (and optionally a volume) to be used for one operation by FATS. "x" is a digit from 1 to 9 and corresponds to the digit on a FATS control statement (e.g., WRITE(1) uses DD statement TAPE1).

The DD statement must specify an appropriate UNIT= parameter to allocate the type of tape drive required for the tapes to be processed. This can be a generic device type (such as UNIT=3590-1) or an installation-defined esoteric name (such as UNIT=TAPE, UNIT=CART).

Any type of label processing may be specified (SL, AL, NL, BLP), but it is frequently desirable with FATS to specify LABEL=(,BLP). If installation conventions do not allow the use of BLP, the FATS parameter BLP can be used to internally change the label type to BLP before open; see Section 11.8 for security considerations for the use of BLP.

WARNING: For proper operation of FATS, the TAPEx DD statement must contain a DSN= parameter and it must not specify a temporary (&&) name. Also the parameters DISP=(...,PASS) or VOL=(...,RETAIN) must not be specified. Otherwise, the tape may forward space to the end of tape or other errors may occur.

For output volumes (WRITE, LABEL, ERASE):

The TAPEx DD statement will usually say DISP=(NEW,KEEP). A volume serial (VOL=SER=) can be specified to request a specific volume, or omitted to request a scratch volume allowing the operator to mount any appropriate volume.

The volume serial can also be omitted when FATS is supplying the serials internally (via its own VOL= parameter). In this case, the JCL parameter DEFER (as in UNIT=(xxxx,,DEFER)) should be used so that the operating system will not call for a tape mount before FATS fills in the proper volume serial.

NOTE: If a volume serial if specified, the label type is SL, and the FATS MULT parameter is given, for subsequent mounts (after the first) the volume serial will be blanked to request scratch tapes. If this wasn't done, z/OS would just reuse the tape already mounted and process it over and over.

Examples:

```
//TAPE1   DD  DSN=FATS.OUT1,UNIT=3590,
//        DISP=(,KEEP),LABEL=(,BLP)
//TAPE2   DD  DSN=DUMMY,UNIT=(TAPE,,DEFER),
//        DISP=(NEW,KEEP)
```

For input volumes (READ, ANALYZE):

DISP=(OLD,KEEP) should be specified, and a data set name (DSN=), unit (UNIT=) and volume serial (VOL=SER=) must be given (if the FATS parameter VOL= is being used to supply serial numbers, the volume serial in JCL may be any dummy serial, e.g., VOL=SER=DUMMY). If LABEL=(,BLP) is specified, the data set name and volume serial do not have to be the actual values.

If the FATS parameter VOL= is used to supply volume serials, the JCL parameter DEFER (as in UNIT=(xxxx,,DEFER)) should be used so that the operating system will not call for a tape mount before FATS fills in the proper volume serial.

Examples:

```
//TAPE1   DD  DSN=INPUT.DSN,UNIT=3490,
//            VOL=SER=901234,DISP=(OLD,KEEP)
//TAPE2   DD  DSN=DUMMY,UNIT=(TAPE,,DEFER),
//            VOL=SER=DUMMY,DISP=OLD,LABEL=(,BLP)
```

TAPExOUT DD STATEMENT

TAPExOUT (where "x" is the same 1 digit number as in TAPEx above) is used only for the FATS ANALYZE(x) function, which causes the companion product FATAR to be invoked. If present, TAPExOUT will request FATAR to copy the tape mounted on TAPEx. See FATAR documentation elsewhere in this manual for details.

SYSIN DD STATEMENT

The SYSIN DD Statement is the source of FATS control statements. It is normally a "DD *" spool file, but can be any disk or tape file with DCB characteristics RECFM=FB and LRECL=80.

13.1 OPERATOR CONTROL

OPERATOR CONTROL

FATS will terminate by itself when all requested operations have completed. However, sometimes you may wish to terminate FATS early (especially when processing a long list of tape volumes) or to terminate processing of a particular tape volume. There are 2 ways to do this:

- By default, FATS issues a WTOR to the operator's console (message FATSW02). This message can be left outstanding and will delete itself when FATS terminates. However, you may reply to it at any time to terminate FATS or terminate processing of a tape volume. The WTOR can be suppressed by the WTOR=NO operand.
- If the optional MODIFY=YES parameter is specified, the z/OS console commands STOP (P) and MODIFY (F) can be used for the same purposes.

WTOR RESPONSES

The FATSW02 message can be replied to at any time with one of these replies ("nn" is the WTOR reply number associated with the message):

R nn,EOJ - FATS will complete processing of all tapes currently in progress, then will terminate. If any tape mounts are pending, they must be satisfied (those tapes will be processed before termination).

R nn,KEOJ - FATS will terminate immediately with a U0888 abend or return code 12. Tapes in progress will be immediately halted.

R nn,Kuuuu - "uuuu" is the z/OS 3- or 4-digit hexadecimal device address of a tape drive currently in use by FATS. The tape volume being processed on that device will be immediately halted; however, another tape may be mounted on that drive if called for by the FATS control statements. Other drives in use by FATS will not be affected.

STOP/MODIFY COMMANDS

If MODIFY=YES was specified, the z/OS console commands STOP (abbreviated P) and MODIFY (abbreviated F) can be used for the same purposes. In the examples below, "id" is the job name of the FATS job or started task:

P id - stops FATS immediately and is equivalent to the KEOJ reply above.

F id,QUIT - terminates FATS after all tapes that are currently in progress have completed and is equivalent to the EOJ reply above.

F id,Kuuuu - stops processing on the specified tape unit (3- or 4-digit hexadecimal device address "uuuu") and is equivalent to the Kuuuu reply above.

14.0 FATS CONTROL STATEMENTS

14.1 GENERAL

GENERAL RULES All FATS control statements must be input on 80-character records. Only columns 1 to 71 are used by FATS. Each control statement contains:

OPERATION KEYWORD Identifies the operation to be performed by FATS. It can start in any column from 1 to 50 on the input record, but any columns before it must be blank. Only one operation keyword can be specified for each tape drive used by FATS. Valid operation keywords are:

DEFAULT	-- Set defaults for other operations
WRITE	-- Certify a tape
READ	-- Verify readability of a tape
ERASE	-- Erase data from a tape
LABEL	-- Write standard labels on a tape
ANALYZE	-- Invoke FATAR to read/copy a tape

TAPE NUMBER Immediately follows the operation keyword (no intervening spaces) and is enclosed in parentheses. It must be a single digit from 1 to 9 and identifies to which TAPEx DD Statement this operation applies (e.g., "(1)" indicates DD Statement TAPE1). Each digit may appear on only one control statement in a given FATS execution. The tape number is not used on the DEFAULT control statement, but is required on all other control statements.

OPERANDS Are separated from the operation keyword (and tape number) by one or more spaces. Each control statement may have one or more operands (separated by commas) or none at all. The operands modify the function of the operation keyword; they control labeling, tape mode, error retries, etc. The last operand must be followed by a space.

CONTINUATION Operands can be continued onto another control statement by following the last operand with a comma and a space, and continuing the operands in any column of the next statement.

COMMENTS May appear on any FATS control statement. They must be separated from the last operand by at least one space (a control statement containing no options cannot contain comments). Any control statement with an asterisk ("*****") in column 1 will be bypassed by FATS and treated entirely as comments.

14.2 OPERATION KEYWORDS

DEFAULT Used to change the defaults of FATS operands. It affects any other operation keywords that follow it. If several operations are to be used in a FATS step, and they all require one or more of the same operand specifications, DEFAULT is a simple way to enter the options only once. Operands specified on DEFAULT can still be overridden on a specific operation statement.

WRITE(n) Used to certify the quality of new or old scratch tapes by writing special patterns on the entire length of the tape and checking for write errors. **This will destroy any existing data on the tape, so the tape must be in scratch status.**

READ(n) Used to verify the readability of existing data on a tape. READ(n) can read one or more data files, but cannot read past the end of currently recorded data since FATS has no way of detecting the end-of-tape marker in read mode. Can be used on non-scratch tapes since it will not destroy any data.

ERASE(n) Invokes the "Data Security Erase" hardware function to **erase all data** from a tape. It may be used to totally erase a tape, or to erase data past currently existing files. Unlike WRITE(n), the "Data Security Erase" does not involve the tape control unit, so other FATS operations or tape I/O from other jobs can proceed without contention.

WARNING: ERASE will erase all data from a tape or cartridge with no possibility of recovery. Use with caution.

LABEL(n) Writes an IBM standard volume label and data set labels on a scratch tape. LABEL(n) can be used to label new tapes or to re-initialize old tapes. **It does destroy any data on the tape.** Can also create unlabeled tapes.

ANALYZE(n) Invokes FATS' companion product FATAR to read and optionally copy a tape. Complete information on FATAR is contained in other sections of this manual. Although FATAR can be executed as an independent program, execution under FATS allows multiple FATAR tasks to execute in one job step and allows the use of FATS parameters for the selection of tapes to be processed. Your installation must be licensed for FATAR to use this operation.

14.3 OPERANDS

All operands can appear on all operation statements, but some actually apply only to one or more of the operation keywords. They are grouped together by their functional type (as shown in the section headings below), but parameters that apply only to one or more operations are marked as such. All option parameters (except VOL=) can appear on a DEFAULT Statement.

14.3.1 RETRY/ERROR OPERANDS

ERRCLEAN=n Applies to: WRITE, READ

If more than "n" permanent or temporary errors are encountered on a tape, FATS will request the operator to clean the tape drive, and then re-certify or re-verify that tape. "n" may be from 1 to 32767. This cleaning action will be requested only once per tape drive.

Default: 32767

MAXCERR=n Applies to: WRITE, READ

If contiguous permanent errors totaling more than "n" inches are encountered on a tape, FATS will terminate processing that tape. "n" may be from 1 to 32767.

Default: 200

MAXERR=n Applies to: WRITE, READ, ANALYZE

If more than "n" permanent errors are encountered on a tape, FATS will stop processing that tape. "n" may be from 1 to 32767.

Default: 200

MAXTERR=n Applies to: WRITE, READ, ANALYZE

If more than "n" temporary errors are encountered on a tape, FATS will stop processing that tape. "n" may be from 1 to 32767.

Default: 200

RETRY=h
RETRY=(l,h) Applies to: WRITE, READ, ANALYZE

"h" defines number of retries required before an error is considered permanent, and "l" the number of retries before an error (temporary or permanent) will be printed in the detail report. "h" may have values from 1 to 99, and "l" may go from 0 to 99 but must be less than "h". The value of "h" affects the summary report: for 10 or less the summary report will show temporary and permanent errors by number of retries; for over 10 all temporary and permanent errors will be totaled.

Default: (1,10)

STOP Applies to: WRITE, READ

When this option is specified, FATS will stop and issue message FATSW05 to the system operator whenever a permanent error is encountered, allowing inspection of the tape. After inspection, the operator may reply to the message requesting continuation (and stopping at the next error), termination, or continuation without stopping. Physical inspection of the tape may not be possible on all types of tape drives.

Default: Do not stop on errors.

STOPNUM=n Applies to: WRITE,READ

Invokes the STOP option (above) after "n" permanent errors have been encountered, allowing the first "n" errors to be skipped without operator interaction. Do not specify STOP if STOPNUM=n is specified.

Default: Do not stop on errors.

14.3.2 VOLUME MOUNTING OPERANDS

BLP Applies to: ALL

The BLP operand is provided to help when installation conventions do not allow LABEL=(,BLP) to be specified on a JCL DD statement, but it desired to bypass label processing on FATS tapes. BLP processing may be required when certifying/labeling virgin or unknown tapes, or when reading multiple input tapes with varying data set names. When BLP is specified, the JCL label parameter is internally changed to BLP before the tape is opened. See Section 11.8 for security considerations for the use of BLP.

Default: Honor the JCL LABEL= parameter.

MAXCLEAN=n Applies to: ALL

When multiple tapes are being processed on a tape drive, specifies that the system operator is to be requested (via console message FATSW06) to clean the tape drive after every "n" tapes are completed. This is to prevent dirt or oxide deposited on the read/write heads by previous tapes from causing spurious errors.

Default: No cleaning requested between tapes.

MAXVOLN=n Applies to: ALL

Limits the number of volumes that FATS will request on a given tape drive when either the MULT or VOLINCR=n operands are used.

Default: No limit for MULT or 682 for VOLINCR=n.

MULT
NOMULT Applies to: ALL

MULT requests that multiple tapes be processed. As each tape is completed, a mount for a new tape will be issued. This will continue until the operator terminates FATS (see Section 13.1), or until the number of tapes specified by the MAXVOLN=n parameter have been processed. NOMULT (the default) is used to turn off MULT if it has been specified on a DEFAULT statement.

Default: Process only one tape, unless VOL= specified.

VALIDATE= Applies to: WRITE, ERASE, READ, ANALYZE

If label processing is bypassed because of the FATS BLP operand or LABEL=(,BLP) on TAPEx DD statements, tape management will not verify the volume serial or expiration date of the tape, so it is possible that the wrong tape may be accidentally mounted and overwritten. However, if VALIDATE= is specified, FATS will verify the volume serials of the mounted tapes for input tapes (VALIDATE=INPUT), output tapes (VALIDATE=OUTPUT), or both (VALIDATE=ALL). The volume serial compared is the serial in the TAPEx DD statement or that provided by the FATS VOL= parameter. For output tapes, the expiration date in the tape labels will also be checked. If the wrong volume serial is mounted, or if the expiration date is not yet reached, FATS will issue message FATSW08 to the system operator giving the option of ignoring the error, mounting another tape, or skipping the tape. For WRITE and ERASE operations, validation will be done only if the SAVLAB operand is specified or defaulted. VALIDATE= will be ignored if the NOREWIND operand is given.

Default: No validation.

VOL=v
VOL=(v,..,v)

Applies to: ALL (cannot appear on a DEFAULT statement).

Specifies one or more tape volume serials to be processed by the operation statement on which it appears. "v" is a 1-to-6 character volume serial. VOL=v requests one volume. VOL=(v,v,...,v) requests multiple volumes; in this format, up to 682 serials may be supplied; the serial list may be continued onto multiple input statements by placing a space after any comma in the volume list, and continuing the list in any column of the next input record. When used in conjunction with the VOLINCR=n and MAXVOLN=n parameters, VOL= can be used to request a sequence of tapes without actually entering all of their serials. Note that VOL= on a LABEL statement implies that FATS will label the tapes with the given volume serial; on a WRITE or ERASE statement, this is true only if the NOSAVE parameter is also specified.

Note: Specifying VOL= makes MULT the default processing.

VOLINCR=n

Applies to: ALL

This operand can be used only in conjunction with the VOL= operand. When specified, the last (or only) volume serial specified by VOL= will be incremented by "n" which must be a 1-to-6 digit decimal number. That volume serial must end in at least one numeric digit; only the trailing numeric part of the serial is incremented. The serial will be repeatedly incremented by "n" until either 682 volume serials have been generated or until the serial cannot be incremented further without "overflowing" the numeric portion. Although a large number of such serials may be generated by VOLINCR=, the MAXVOLN=n operand may be used to limit the number of volumes that will actually be requested by FATS. VOLINCR=1 may be used to cause a number of consecutively numbered tapes to be requested.

Default: No incremented serials are generated.

14.3.3 LABELING OPERANDS

ANSI Applies to: WRITE, LABEL, ERASE

Requests that tapes are to be labeled with ANSI V3 labels instead of IBM standard labels.

Default: IBM standard labels will be written

DD=LABEL Applies to: WRITE, LABEL, ERASE

Causes the tape to be labeled with the volume serial specified on the associated TAPEx DD statement. Cannot be used with the MULT parameter, and also should not be used when VOL= has been specified on the WRITE parameter.

LABEL=vvvvvv Applies to: WRITE, LABEL, ERASE

Causes the tape to be labeled with volume serial "vvvvvv". Cannot be used with the MULT or VOL parameters, and also should not be used when VOL= has been specified on the WRITE parameter.

NOLABEL Applies to: WRITE, ERASE, LABEL

Resets all LABEL options (including SAVLAB). The tape being processed will become an unlabeled tape. When used with the LABEL(n) statement, NOLABEL will quickly initialize unlabeled tapes.

OPERATOR
NOOPERATOR Applies to: WRITE, LABEL, ERASE

If OPERATOR is specified, as each tape is opened, the system operator will receive message FATSW01 and must reply with the volume serial to be written on the tape. NOOPERATOR is used only in conjunction with SAVLAB (specified or by default) to prevent SAVLAB from going into OPERATOR mode if a tape is not labeled (see below), so that unlabeled tapes will remain unlabeled.

OWNERID='....' Applies to: WRITE, LABEL, ERASE

Specifies the contents of the owner ID field in volume labels written by FATS. 1 to 10 characters (including blanks) may appear between the apostrophes. OWNERID= has an effect only if FATS is writing a volume label due to other labeling options. Even if SAVLAB is used, OWNERID= may be used to change the owner ID.

Default: 10 spaces (or original value if SAVLAB)

SAVLAB
NOSAVE Applies to: WRITE, ERASE

If SAVLAB is specified or defaulted, and the tape being certified contains IBM standard labels, the tape will be re-labeled with its original volume serial; if the tape does not contain IBM standard labels, the tape will be labeled with a serial provided by the system operator (see OPERATOR above) unless the NOOPERATOR parameter is present. NOSAVE will prevent the volume serial from being preserved. Note that SAVLAB preserves only the volume serial and owner ID from the volume label; the contents of data set header labels is not saved.

Default: SAVLAB

14.3.4 PRINT OPERANDS

LINECNT=n Applies to: ALL

Specifies the number of lines per page to be printed on FATS reports. If specified on a DEFAULT statement, it will control the page size of the FATS control report and summary report as well as providing a default for the detail reports. If specified on any other operation statement, it modifies only the detail report page size for that operation.

Default: 60 lines per page

MAXDETCNT=n Applies to: ALL (except ANALYZE)

Specifies the maximum number of detail report lines that will be printed together before double spacing. This will be effective only if a separate detail report DD statement SYSPRINx has been provided.

Default: 4

NEWPAGE Applies to: ALL (except ANALYZE)

When multiple tapes are processed on a tape drive, the NEWPAGE operand requests that the detail report will skip to a new page whenever a new volume is processed, making the report easier to separate by volumes. This will be effective only if a separate detail report DD statement SYSPRINx has been provided.

Default: Skip a single blank line between volumes.

NONMETRIC
METRIC Applies to: ALL

When NONMETRIC is specified (or defaulted) FATS will list tape lengths and error positions and lengths in U.S. units (feet and inches). When METRIC is specified FATS will list tape lengths and error positions and lengths in metric units (meters and centimeters).

Default: NONMETRIC.

THRESHOLD=n Applies to: WRITE, READ

Specifies the maximum number of permanent and temporary errors that will be printed in the detail report for any given tape before printing is suspended. The summary report will still recap the total number of errors. This is used to limit the size of the printout.

Default: 1000 errors.

14.3.5 CONTROL OPERANDS

See Section 13.1, Operator Control, for more information on the use of these operands.

MODIFY=YES
NO

Applies to: DEFAULT

Specifies whether FATS will respond to an operator's MODIFY (F) and STOP (P) console commands. FATS ordinarily issues a WTOR to the console at startup that operator can reply to at any time to control FATS execution (see the WTOR parameter). If MODIFY=YES is specified the operator may use the MODIFY (F) and STOP (P) commands to control FATS processing.

Default: MODIFY=NO

WTOR=YES
NO

Applies to: DEFAULT

Specifies whether FATS will issue its WTOR message (FATSW02) in order to provide operator control over FATS processing. If MODIFY=YES is specified to allow operator control with OS commands, it is not necessary to have the outstanding reply available to control processing.

Default: WTOR=YES

14.3.6 MISCELLANEOUS OPERANDS

BLKSIZE=n Applies to: WRITE

Specifies the size of the block that will be written to tape during certification, in bytes. "n" must be less than 262145, and more than twice the density (BPI) of the tape being certified.

Default: Device-dependent; 262140 (256K-4) bytes for most cartridge tapes when large blocks are supported by the system, and 65535 (64K-1) bytes when large blocks are not supported.

FILES=n Applies to: READ

"n" specifies how many physical files FATS is to read from its input tape. A physical file is defined as all records up to and including one tape mark. If the input tape contains labels, the labels before and after each data file will each count as one file, so each labeled file is actually three files. If NOREWIND is specified, the header label file of the first data file will not be read. If "n" is 0, FATS will terminate when it reads two consecutive tape marks (with no intervening data); this is usually sufficient to read all data files on the tape.

Default: 3 files (one labeled file).

WARNING: If "n" specifies more files than are present on the tape, or if "n" is 0 and the tape does not end in two tape marks, the tape may physically run off the end of the reel and require operator intervention to rethread it (tape cartridges do not have this problem, but will get an error instead). Multi-volume data sets do not have double tape marks at the end except for the last volume. FATAR, the companion product to FATS, can properly read multi-volume tapes and is a better choice for most verification operations.

NUMWRITES=n Applies to: WRITE

Specifies the number of WRITE CCWs that will be chained together during a certify operation. In other words, this is the number of test blocks that will be written in one I/O. Increasing the value may improve certification performance but may impact other jobs using tape drives on the same string. The value can be from 1 to 50.

Default: 20

RESIDUAL Applies to: ERASE

Causes FATS to locate the end of the last existing file on the tape and erase the remainder of the tape. For 3480 tapes, FATS will read analyze the tape to find the last file (for unlabeled tapes, it locates a double tape mark). For other cartridge drives, it locates the EOD mark written by the drive after the last file. This is used to erase residual data from a non-scratch tape containing data files.

Default: Erase the entire tape.

RETCODE Applies to: ALL

Causes FATS to terminate with a return code of 12 rather than abending in all cases where it would terminate with a U0888 abend code (due to major errors).

Default: Abend with U0888 if major errors occur.

REWIND
NOREWIND

Applies to: ALL

REWIND causes FATS to rewind all tapes to load point before starting the specified operation, regardless of any LABEL= parameter in the JCL. NOREWIND will tell FATS to leave the tape at the user specified position; for WRITE and ERASE statements, this can be used to certify/erase the remainder of a tape beyond existing files. NOREWIND resets all label options.

Default: REWIND

WTO

Applies to: ALL

Causes FATS to report the results of each operation on the system console via a one line message indicating the total number of permanent and temporary errors, or abnormal completion of the operation.

14.3.7 ANALYZE OPERANDS

All of the operands that are valid on an ANALYZE statement input to FATAR (as described later in this manual), are valid on a DEFAULT statement or an ANALYZE(n) statement when FATAR is invoked as a subtask of FATS.

16.0 EXAMPLES OF FATS USAGE

INTRODUCTION

Examples are provided here to guide you in the use of FATS. As many of the common uses of FATS as possible have been included, and most of the control statements and parameters are illustrated here. However, be aware that they are just examples and must be customized to your installation and situation before use. Areas of such customization will probably include: data set names, unit names for tape, special parameters for tape management systems. The examples assume that no special STEPLIB DD statement is required to execute FATS; this may not be true in your installation.

Because of the many possible combinations of operands, control statements, tape label types, etc., it is not practical to create examples that cover all potential FATS usage. Many examples illustrate more than one aspect of FATS use, e.g., two or more operands together. This does not mean that the two must always go together unless the accompanying text says so.

Many examples use LABEL=(,BLP) (bypass label processing) on the tape DD statements. For input tapes, BLP allows you to mount a labeled tape without knowing its volume serial or data set name; for output, it allows creation of unlabeled tapes. Sometimes this is simply a convenience, but for many FATS operations, such as labeling and certifying new tapes, it is essential. In some installations, system parameters prevent the usage of BLP by most users; in this case, the FATS control statement parameter BLP can be substituted. See Section 11.8 for security considerations for the use of BLP.

ISPF DIALOG SUPPORT

Most of the FATS functions can be performed by using the FATS ISPF interface; see Section 93.1 for further details.

INDEX TO EXAMPLES

The first-time or infrequent user of FATS should review many of the examples below since they frequently build upon one another in illustrating the usage and effects of various parameters. For your convenience, however, here is a list of the examples provided to aid in quickly finding the one you need. It shows the title of the example and the operations and keywords it illustrates.

#	EXAMPLE NAME	OPERATION	KEYWORDS
1	Certify SL tapes	DEFAULT WRITE	MULT,MODIFY
2	Certify SL tapes on multiple drives	WRITE	VOL
3	Certify and label tapes	WRITE	VOL,VOLINCR,NOSAVE, MAXVOLN,MAXERR, MAXCERR,NEWPAGE
4	Certify scratch tapes on 3 drives	DEFAULT WRITE	MULT,MAXVOLN, MAXCLEAN, METRIC
5	LABEL tapes	DEFAULT LABEL	LINECNT,OWNERID,VOL, VOLINCR,MAXVOLN,BLP
6	Label and certify new tapes	WRITE	VOL,VOLINCR,MAXCLEAN, WTO, NOSAVE
7	Verify a SL tape	ANALYZE	
8	Verify a multi-file SL tape	READ	VOL,FILES
9	Verify on several drives	DEFAULT READ	VOL,FILES
10	Erase on multiple drives	DEFAULT ERASE	VOLINCR,MAXVOLN,VOL
11	Erase residual data	ERASE	RESIDUAL,VOL
12	Erase multiple tapes with any volume serials	ERASE	MAXVOLN

16.1 EXAMPLES

EXAMPLE 1: CERTIFY SL TAPES

This example certifies any number of standard labeled tapes. The UNIT= parameter controls which type of tape drive is allocated for certification (e.g., 3480, 3490, 3590-1). Since no volume serial is specified, the operating system will ask for a scratch tape. With LABEL=(,SL) specified in the JCL, tapes with any volume serials can be mounted. A tape management system, if present, can protect against accidentally overwriting active data by confirming the scratch status of any tape mounted. The MULT parameter will cause FATS to continuously request new scratch tapes until the system terminates FATS (See Section 13.1). Since the SAVLAB parameter is the FATS default, the original volume serial of each tape will be preserved. The PARM=I on the EXEC statement causes FATS internal documentation to be printed on SYSPRINT.

```
//FATS     EXEC PGM=FATS,PARM=I,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=TAPE,
//            LABEL=(,SL),DISP=(,KEEP)
//SYSIN    DD *
    DEFAULT MODIFY=YES
    WRITE(1)MULT
```

EXAMPLE 2: CERTIFY SL TAPES ON MULTIPLE DRIVES

Certify 3 standard labeled tapes on two tape drives, specifying the required volume serials in FATS control statements via the VOL= parameter. UNIT=(3490,,DEFER) on the TAPE DD statements will prevent the operating system from requesting a mount until FATS passes it the required volume serial. This example certifies 3490E cartridges, but any valid tape device type can be substituted. Each tape must be previously labeled with the proper volume serial since FATS is opening them as labeled. Since the SYSPRIN1 and SYSPRIN3 DD statements are present, separate detail reports will be printed for each tape drive. Since the SUMMPRT DD statement is present, a separate summary report for all tapes on all drives will be printed.

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SYSPRIN3 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FAT1,UNIT=(3490,,DEFER),DISP=(,KEEP)
//TAPE3    DD DSN=FAT2,UNIT=(3490,,DEFER),DISP=(,KEEP)
//SYSIN    DD *
    WRITE(1) VOL=111111
    WRITE(3) VOL=(222222,333333)
```

EXAMPLE 3: CERTIFY AND LABEL TAPES

Twenty tapes, with volume serials "AX0001" to "AX0020" are to be labeled and certified. Their current internal volume labels are unknown so label processing must be bypassed (LABEL=(,BLP) in JCL). VOL=AX0001 provides the first volume serial to be requested, VOLINCR=1 specifies that the serial is to be incremented by 1 each time, and MAXVOLN=20 limits FATS to 20 such volumes. The presence of VOL= and NOSAVE causes the new tapes to be re-labeled with the AXnnnn volume serials. NEWPAGE causes the detail report for each tape to start on a new page (for easier separation), and MAXERR=5 and MAXCERR=3 will cause any given tape to terminate if it has more than 5 permanent errors or a 3 inch continuous error.

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE5    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//SYSIN    DD *
    WRITE(5) VOL=AX0001,VOLINCR=1,NEWPAGE,NOSAVE,
      MAXVOLN=20,MAXERR=5,MAXCERR=3
```

EXAMPLE 4: CERTIFY SCRATCH TAPES ON 3 DRIVES

94 scratch tapes are to be certified on three Magstar (3590) tape drives. The system operator knows which tapes to mount, so FATS is to simply call for scratch tapes. The DEFAULT statement is used so that the other FATS parameters only need to be specified once. MULT will cause FATS to repeatedly call for new tapes on each drive (since no volume serial is given in the JCL, scratches will be requested). MAXVOLN=31 causes each drive to terminate when 31 tapes have been certified on it (for the third drive, 32). MAXCLEAN=10 instructs FATS to ask the operator (via a console message FATSW06) to clean each tape drive between every set of 10 tapes (i.e., 3 cleanings per drive in this example) so that dirty tape drives don't cause false errors. Lengths will be reported in meters and centimeters.

***Note**:* The SAVLAB parameter, which is the default, causes FATS to preserve the volume serials on the tapes; if any tape does not contain a label, FATS will ask the operator for the correct label.

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SYSPRIN2 DD SYSOUT=*
//SYSPRIN3 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=(3590-1,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//TAPE2    DD DSN=FATS,UNIT=(3590-1,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//TAPE3    DD DSN=FATS,UNIT=(3590-1,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//SYSIN    DD *
   DEFAULT MULT,MAXVOLN=31,MAXCLEAN=10,METRIC
   WRITE(1)
   WRITE(2)
   WRITE(3) MAXVOLN=32
```

EXAMPLE 5: LABEL TAPES

Volumes "TST990", "TST994", and "TST998" are to be labeled with those volume serials. The VOL= parameter specifies the first of those serials. The VOLINCR=4 parameter causes that serial (TST990) to be incremented by 4. Although MAXVOLN=50 is specified, the incrementing stops when the numeric portion of that serial cannot be incremented without overflowing. LINECNT=80 requests 80 lines per page on all FATS reports.

The BLP operand was specified on the DEFAULT statement. This can be used in installations where the LABEL=(,BLP) operand is treated as LABEL=(,NL) due to installation options. The FATS BLP operand forces BLP (bypass label processing) internally at OPEN time), overriding the JCL specification. See Section 11.8 for security considerations for the use of BLP.

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//SYSIN    DD *
    DEFAULT LINECNT=80,OWNERID='TEST TAPE',BLP
    LABEL(1) VOL=(TST990),
         VOLINCR=4,MAXVOLN=50
```

EXAMPLE 6: LABEL AND CERTIFY NEW TAPES

The tape librarian has 250 brand new tapes to label and certify (certifying is recommended to clean loose oxide from the new tapes) starting at volume serial 500001. The VOL= and VOLINCR= parameters will cause this to happen. Since no MAXVOLN= parameter was specified, FATS will call for up to 682 volume serials (up to 500682). However, when all tapes are done, or if the certification process must be stopped to do other work, the librarian can stop FATS from the z/OS console (see Section 13.1). The WTO parameter will cause a 1-line summary of the errors on each tape to be written to the console so that potential problem tapes can be immediately identified. Since these are new tapes, the MAXCLEAN=5 parameter requests tape drive cleaning after every five tapes to remove accumulated oxide.

NOSAVE prevents reading blank tape and allows labeling to take place.

***Note**:* To simply label the tapes without certification, change WRITE(1) to LABEL(1).

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//SYSIN    DD *
    WRITE(1) VOL=500001,VOLINCR=1,MAXCLEAN=5,WTO,NOSAVE
```

EXAMPLE 7: VERIFY A SL FILE

A standard labeled tape "222222" is to be verified for readability. The FATS companion product FATAR is to be used to do so. Since FATAR is opening it as labeled, the proper data set name and volume serial must be given. Since the FILES= parameter is not given, FATAR will read to the end of the tape (2 tape marks or EOV labels).

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN7 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE7    DD DSN=DATA.SET.NAME,UNIT=TAPE,
//            VOL=SER=222222,DISP=OLD
//SYSIN    DD *
  ANALYZE(7)
```

EXAMPLE 8: VERIFY A MULTI-FILE SL TAPE

A standard labeled tape "333333" containing two files is to be certified for readability. The volume serial is specified via the FATS parameter VOL=, but a dummy serial must be given on the TAPE DD statement to satisfy operating system requirements; UNIT=(TAPE,,DEFER) keeps the system from requesting the mount until FATS sets the actual serial. LABEL=(,BLP) is specified since the true data set name of the first file is unknown. FILES=6 causes two standard label files to be read (the header and trailer label files count as files to FATS).

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=DATA.SET.NAME,UNIT=(TAPE,,DEFER),
//            VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//SYSIN    DD *
  READ(1) VOL=333333,FILES=6
```

EXAMPLE 9: VERIFY ON SEVERAL DRIVES

Several tapes are to be verified for readability using two tape drives. The VOL= parameter specifies which tapes are to be mounted on each drive. FILES=0 tells FATS that each tape is to read until two consecutive tape marks are encountered (two tape marks normally indicates end of data on the tape).

WARNING: Tapes that are part of a multi-volume set (except the last) DO NOT end in two tape marks and may run off the end of the reel if read with FILES=0. FATAR is recommended to verify such tapes.

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SYSPRIN2 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=ANY.NAME,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),VOL=SER=DUMMY1,DISP=OLD
//TAPE2    DD DSN=ANY.NAME,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),VOL=SER=DUMMY2,DISP=OLD
//SYSIN    DD *
    DEFAULT FILES=0
    READ(1) VOL=(111111,222222,333333)
    READ(2) VOL=(444444,555555,666666)
```

EXAMPLE 10: ERASE ON MULTIPLE DRIVES

A large number of tapes are to be "Data Security Erased" before they are discarded. Since they have consecutive serial numbers, the FATS parameters VOL=, VOLINCR=1, and MAXVOLN=50 are used to erase 50 tapes on each of 3 tape drives. FATS will call for the tapes by serial as needed. Although LABEL=(,BLP) has been specified to suppress operating system label handling, FATS will re-label each tape as it erases it. Note that "Data Security Erase" operations do not tie up the tape control unit, so that each of the three erase operations can proceed at full tape speed, and other tape jobs will not be impacted. **Innovation strongly recommends that you run ERASE on a tape before writing data to be sent to a third party to insure that all sensitive data left from previous use of the tape is removed.** (Before the ERASE, some users run a CERTIFY on the tape for additional assurance that the data will not be readable after the ERASE.)

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SYSPRIN3 DD SYSOUT=*
//SYSPRIN5 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//TAPE3    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//TAPE5    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//SYSIN    DD *
    DEFAULT VOLINCR=1,MAXVOLN=50
    ERASE(1) VOL=100001
    ERASE(3) VOL=100051
    ERASE(5) VOL=100101
```

EXAMPLE 11: ERASE RESIDUAL DATA

Several tapes that contain data are to be sent off site, but you want to be sure that any residual data on those tapes, from previous uses of the tape, is erased. This insures that sensitive residual data can't be recovered from the tapes.

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,BLP),DISP=(,KEEP)
//SYSIN    DD *
    ERASE(1) RESIDUAL,VOL=(005279,013424,025993)
```

EXAMPLE 12: ERASE MULTIPLE TAPES WITH ANY VOLSERS

You want to create a job which will erase any 20 tapes (MAXVOLN=20), without specifying the volume serials of the tapes to be erased. (You can specify any number of tapes instead of "20".) Whichever 20 tapes you put into your tape loader will be the ones which will be erased. You can add new tapes to the loader at any time by taking out the tapes that have already been erased.

If you are erasing a large number of tapes in a single job, you may want to add a cleaning cartridge to the tape loader so that the drive gets cleaned during the job to avoid problems which may arise from a dirty drive. The drive will recognize that this is a cleaning cartridge, and it won't interfere with the Erase job (the cleaning cartridge won't be counted as one of the "n" tapes you want to erase).

```
//FATS     EXEC PGM=FATS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=FATS,UNIT=(TAPE,,DEFER),
//            LABEL=(,SL),DISP=(,KEEP)
//SYSIN    DD *
    ERASE(1) MAXVOLN=20,MULT
```

This page intentionally left blank.

20.0 FATAR FUNCTIONAL DESCRIPTION

WHAT IS FATAR?

FATAR (Fast Analysis of Tape And Recovery) is a multi-purpose tape utility, that can be executed under z/OS. FATAR can read any magnetic tape that is usable by that operating system, and can process multiple files and multiple volumes in one execution. It offers the ability to summarize or examine in detail the data read from the tape. FATAR's error recovery procedures are often capable of reading data that causes data check errors when read by normal access methods.

FATAR can also create a copy of the input data on another tape, optionally applying user-specified modifications as it copies. FATAR never modifies the input tape, so creating a copy is the only way to recover from errors on the original input. FATAR can also be used to move data from physical tapes to virtual tape systems.

In many installations, much of the installation's data is stored on magnetic tape, which is subject to many hazards. This data will be useful only if it was written correctly when first recorded and is still readable when needed. Many hours of staff and computer time can be wasted trying to recover from physical (hardware) or logical (data) errors on tape. FATAR can greatly reduce this recovery time, or even discover problems before they occur.

Innovation's FATSCOPY program is the preferred method for copying tapes, especially if a large number of volumes need to be copied to new media. In addition to providing better performance, FATSCOPY propagates tape management information from the input data set to the output data set for IBM's RMM and CA Technologies' CA 1 and TLMS. The FATSCOPY panels and keywords allow you to control which volumes and data sets are copied and how they are grouped on the output tapes. FATSCOPY also provides checkpoint, restart, and audit facilities. Refer to Section 30 of this manual for more information about FATSCOPY.

WHO USES FATAR?

FATAR is ideal for use by tape librarians and operations personnel, by applications programmers, and by systems programmers.

WHAT IS FATAR USED FOR?

FATAR has many applications, including:

- Investigating an "unknown" tape, to discover its label type, file count, DCB characteristics, etc.
- Mapping a tape, providing a compact summary of the characteristics of all files on the tape.
- Examining the data on a tape.
- Verifying that a tape file is properly formatted (every block is checked against its DCB information).
- Detecting and correcting invalid variable spanned records.
- Verifying that certain data fields contain valid data.
- Scanning a tape for read data checks, which may cause application job failures.
- Verifying the readability of critical or archive tapes.
- Correcting invalid data or data checks by creating a copy of the input tape(s) with the bad data corrected or dropped.
- Creating a good copy of a tape that was not properly closed (such as during a system failure).

- Recovering data from a tape partially overwritten in error with a shorter file.
- Creating a backup copy of any tape (or multi-volume set of tapes), even if multiple files exist on the tape.
- Replacing tapes with small blocksizes with copies using a larger blocksize (reducing elapsed and CPU times of applications that read that data).
- Creating a copy of a tape with a different label type from the original (such as AL-to-SL or NL-to-SL).
- Creating copies of tapes at higher density or on a different type of tape drive.
- Copying tapes with blocksizes up to 256K, including ABR backup tapes.
- Making an "image copy" (an exact bit-for-bit copy) of a tape volume.

FATAR CONTROLS

FATAR has a flexible set of command statements entered as 80-character records, of which only columns 1-71 are used by FATAR. For many operations, the commands required are very simple since the default operations of FATAR are designed to handle the most common user requirements. However, the FATAR statements allow you to greatly modify FATAR operation, including the ability to specify an operation down to a specific block and record on the tape if required.

INPUT TAPE PROCESSING

Unless you instruct it otherwise, FATAR will read all files on the tape volume (or multi-volume tape set) specified on its TAPEIN DD statement. If DCB information is available (from JCL or from the tape labels), FATAR will deblock each tape block into individual records if the record format is fixed or variable, allowing you to examine or modify each logical record. If you prefer, or if DCB information is not available, FATAR will treat each tape block as one record.

Since most tapes in IBM installations are labeled, FATAR will normally expect input tapes to contain labels (either IBM standard or ISO/ANSI), and will intelligently print the labels, extract data set name and DCB information from them and treat only real data files as data. This will be true even if you have specified LABEL=(,BLP) in JCL, so that you can mount tapes whose volume serial you don't know or don't care about and still print and process their labels. However, FATAR will recognize unlabeled tapes automatically and treat all files on them as data. If you have a need to process label files as data, you can instruct FATAR to do so.

If the input data is in ASCII, FATAR will translate it to EBCDIC before processing. By default, the ASCII-EBCDIC translation table used includes only the most common characters. If you need translation of other characters, such as the Euro character, see the ASCII= operand for custom translation tables.

DATA BLOCK PROCESSING

Once a block has been read, FATAR offers facilities via FATAR commands to:

- Print the block in EBCDIC, HEXADECIMAL, or DUMP format (HEX and EBCDIC together). All or part of the block may be printed.
- Change the length of the block; extra bytes may be added at the beginning or end of the block, or both.
- Change data in the block, either at a specified location or by scanning for a string.
- Scan specified fields in the block for valid packed or zoned decimal numbers, to help eliminate data exceptions in application programs.
- Scan specified fields in the block for validity by comparison to a user specified table of valid or invalid values.

- Scan for specific strings of data, in specified locations or in any location.
- Drop data (prevent copy to output tape).

If logical records are being deblocked, all of the above (except block length changing) can be applied to each logical record.

NOTE: Changes to the input blocks never affect the input tape. Modified blocks are written only to the output tape, if present.

OUTPUT TAPE PROCESSING

If the output tape DD statement TAPEOUT is present, FATAR will copy each file read from TAPEIN to the output tape. This will be an exact copy of the input, unless you request modification of one or more data blocks, or dropping of certain data blocks. You may also request that entire files be dropped from the copy.

By default, any block that had an unrecoverable data check when read from TAPEIN will not be written to TAPEOUT, but you may override this.

FATAR will issue a standard OPEN and CLOSE for each file on TAPEOUT.

If the output tape is in ASCII format, the data blocks will be translated from EBCDIC to ASCII before they are written. If the records are variable format, appropriate conversion to or from the ASCII variable format and the IBM format will be done. By default, the EBCDIC-ASCII translation table used includes only the most common characters. If you need translation of other characters, such as the Euro character, see the ASCII= operand for custom translation tables. Note that if both the input and output tapes are in ASCII format, data will be copied without translation but any data displayed or tested by FATAR will be translated to EBCDIC.

FATAR PERFORMANCE

FATAR must read every block on the input tape(s) and, when copying, write each block to the output. Its performance is dependent on several factors, including:

- the blocksize of the data file; smaller blocks will take longer
- the type of tape drive
- the type of channel on which the drive is attached (FICON or ESCON)

To give you some idea of what to expect, we have run tests at Innovation using tapes containing large blocksizes. These tests used IBM 3490Es, IBM 3590s, IBM TS1130s, EMC DLms, and StorageTek T10000s on FICON channels. Both the 3490E and 3590 volumes were IDRC-compressed. The 3490E volume contained 1.3 GB of data (before compression) and the 3590 volume contained 43.8 GB (before compression).

1. Copy one 3490E volume (1.3 GB before compression) to one TS1130 volume – 8.0 minutes (2.8 MB/sec).
2. Copy one 3490E volume (1.3 GB before compression) to one T10000 volume – 8.5 minutes (2.6 MB/sec).
3. Analyze one 3590 volume (43.8 GB before compression) – 79.0 minutes (9.5 MB/sec).
4. Copy one 3590 volume (43.8 GB before compression) to one EMC DLm volume – 76.5 minutes (9.8 MB/sec).

These are sample times from a specific environment; your results will vary.

This page intentionally left blank.

21.0 FATAR TECHNICAL SUMMARY

21.1 GENERAL

THE FATAR PROGRAM

Innovation recommends that FATAR (Fast Analysis of Tape And Recovery) be run using REGION=0M.

FATAR **must** be linkedited and executed as an authorized program.

FATAR may also be called as a sub-task of its companion product FATS. This mode of operation allows multiple FATAR sub-tasks to operate in the same job, and also allows use of FATS keywords for specifying multiple executions of FATAR using the same input tape drive, but with different input tapes.

CONTROL STATEMENTS AND REPORT OUTPUT

FATAR accepts its control statement input from DD statement SYSIN. If no SYSIN DD is present in the step, all defaults are assumed. The control statements are 80-byte records, of which only columns 1 to 71 may contain information.

Report output is directed to DD statement SYSPRINT, which may be a SYSOUT file or any tape or disk file valid for output. SYSPRINT always has a logical record length (LRECL) of 121 bytes and writes Fixed Block (RECFM=FB) records. If SYSPRINT is directed to SYSOUT or if the blocksize is unspecified, 121 is assumed. The user may specify a blocksize for tape or disk output, but if it is not a valid multiple of 121, 1210 will be assumed.

SYSPRINT will contain a report identifying the tape or tapes read, including the volume serials, tape and drive type, and density (if applicable). For each file read, the header and trailer labels are displayed (if it is a labeled tape), and the number of records read are shown for each data file. Total data blocks and bytes are displayed for the entire tape. Any blocks that had data checks on the input are identified and displayed. If the input tapes are compacted by IDRC hardware, compaction information is displayed. FATAR control statements can request additional printout, such as printing data blocks or formatting label fields.

If the optional TAPESUMM DD statement is included, it will contain a page for each tape processed, with a 2 line summary of each file read, including information from the tape labels and a summary of the data read. DCB characteristics of TAPESUMM are the same as for SYSPRINT, described above. For compacted input tapes, IDRC statistics are displayed for each file and for the entire tape.

Section 24.8 contains examples of SYSPRINT and TAPESUMM output.

If FATAR is run as a sub-task of FATS, these differences apply:

- Printout will be directed to SYSPRINx, where "x" is the same as the digit on the FATS control statement ANALYZE(x).
- Most ANALYZE operands can be specified on the FATS ANALYZE(x) statement, but the user can override those options or provide Modification/Scan statements by supplying them in a DD statement SYSINx ("x" same as above).

21.2 INPUT TAPE PROCESSING

INPUT TAPE TYPES

The input to FATAR may be any one of the following:

- A standard labeled (SL) tape containing one or more files (data sets).
- A multi-volume set of standard labeled tapes containing one or more files.
- A tape with ISO/ANSI V3 or V4 labels (AL) containing one or more files in ASCII.
- A multi-volume set of ISO/ANSI V3 or V4 labeled tapes containing one or more files in ASCII.
- An unlabeled tape containing one or more files, recorded in EBCDIC or ASCII.
- An unlabeled multi-volume set containing only one file (unlabeled multi-volume multi-file tape sets cannot be supported by FATAR since it cannot tell how many files are on each tape).

FATAR can read tapes on any IBM or IBM-compatible physical tape drive or virtual tape system (VTS).

You can use the FATAR copy function to perform "virtual ejection" by copying a VTS virtual tape to a real tape volume outside the VTS.

INPUT TAPE LABEL HANDLING

Labeled tapes are handled by FATAR independently of the LABEL= parameter in the JCL. In other words, the TAPEIN DD statement may say

```
LABEL=(,SL)  or  LABEL=(,AL)  or  LABEL=(,BLP)
```

and FATAR will still be able to recognize and process the labels on the tape. Standard labels (SL) and ANSI V3 or V4 labels (AL) will be detected even when operating system label processing is bypassed. If ANSI labels are found, it is assumed that the associated data file is written in ASCII, and an ASCII-to-EBCDIC translation will be done on the labels and associated data files.

Unlabeled tapes, mounted with

```
LABEL=(,NL)  or  LABEL=(,BLP)
```

are automatically recognized by FATAR. When it reads the first input block, FATAR sees that it is not a label and automatically treats the tape as unlabeled.

If `LABEL=(,BLP)` is permitted in your installation, it is often useful with FATAR since it allows you to mount a tape whose volume serial or first data set name are unknown, or to set up a standard FATAR procedure that does not need to be modified for every run. Even if BLP is not permitted in JCL, FATAR has an operand BLP that invokes BLP processing internally. See Section 21.6 for security considerations for the use of BLP.

You can force FATAR to process a labeled tape as unlabeled so that all files on the tape (including files containing labels) are handled as data by FATAR, even if `LABEL=(,SL)` or `LABEL=(,AL)` is specified in the JCL; in this case, the tape is positioned so that the first file processed is the header labels of the file specified in the JCL.

If the LABEL= parameter on the TAPEIN DD statement specifies a file other than the first, e.g.,

```
LABEL=5  or  LABEL=(4,BLP)
```

FATAR will start its processing with that file. If the input tape is a labeled tape but you have specified BLP in the JCL, the file number should point FATAR to the header labels of some file on the tape (header labels normally start at files 1, 4, 7, etc.); if you point to a data file instead of a label file, FATAR will treat the tape as unlabeled.

Unless you override normal FATAR operation with the LABELS= operand on the FATAR ANALYZE statement, FATAR will automatically recognize the label type of the first input tape and automatically do the right processing. SL and AL tapes will be automatically identified, the labels will be formatted, and information from the labels will be used to process the input tape (and for creation of files on the output tape, if present). Unlabeled tapes will be automatically recognized and processed.

When reading header labels, FATAR will extract the data set name (only the last 17 characters of the name are recorded in the label), and DCB information (RECFM, LRECL, and BUFOF (ANSI only)). These values will be used for opening the output tape file (if any) and for deblocking the blocks of the following data file. If you have the tape management interface enabled (see Section 21.5), FATAR will query it to get the full 44-character data set names for all files on the input tape, and will use those names for the output files.

When reading trailer labels, FATAR will verify the block count against the actual count of blocks read in the preceding data file, and produce a warning message if they do not match.

If a multi-volume labeled tape set is being read by FATAR, it will perform the processing described above on the EOV labels of at the end of each tape and the header labels at the beginning of the next tape, but will process the file that crossed tapes as one large file.

INPUT FILE HANDLING

For labeled tapes, FATAR will process all of the files on all of the volumes specified on the TAPEIN DD statement unless you deliberately limit it to a certain number of files with the NUMFILES= or ENDAFTER= parameters. FATAR will continue to read and process files until either:

- Two consecutive tape marks are read, with no data or labels between, but only if the two consecutive tape marks are found where header (HDRx) labels would normally be expected.
- The number of data files specified by NUMFILES= has been read.
- The block specified by the ENDAFTER= parameter has been read.
- EOV labels at the end of a tape are read (indicating that the data set is continued on another tape), but no more tape volume serials were specified.

This means that, when in label mode, FATAR can read and copy an entire multi-file, multi-volume tape set automatically.

For unlabeled tapes, FATAR will process multiple files on the input tape, but is unable to switch to new volumes since there is no way to tell how many files are on the current volume, and therefore when to switch volumes. By default, FATAR will process all files on one unlabeled tape volume until two consecutive tape marks are read (as for labeled tapes). FATAR will also stop when it encounters EOV labels (it checks for EOV even though labels are not being processed) since EOV labels are followed by only one tape mark; this prevents FATAR from running off the end of round tapes or getting errors on cartridges. If NUMFILES= was specified, FATAR will stop after reading that many physical files.

For a truly unlabeled tape that is part of a multi-volume set, there is no way for FATAR to detect the end of the recorded data on the tape. In order to avoid an I/O error after the last data (and possibly running off the tape for round tapes), you must tell FATAR how many files exist on the tape with NUMFILES=.

LABELS-ONLY MODE

FATAR supports a mode where only label information is printed from the input tape. LABELS=ONLY operates much like LABELS=YES, except that data files will be rapidly skipped over using a Forward Space File (FSF) CCW. No output tape is allowed. This allows you to quickly display information about all labeled files on a tape volume (or multi-volume set).

OS LABEL MODE

FATAR has a special label mode, LABELS=OS, where all label processing and tape positioning is left to the operating system. Input tapes are handled the same as any utility program or user program, except that they must be on tape. In this mode, FATAR will handle exactly one input file (NUMFILES=1 is forced), but that file can be multi-volume, either labeled or unlabeled, and can even be several files concatenated together, as long as all the files are tape. LABELS=OS can be used to bypass the restriction on multi-volume unlabeled tapes, or to create one output tape file from multiple input files.

IMAGE COPY MODE

FATAR's special image copy mode, LABELS=IMAGE, is used to make an exact bit-for-bit copy of a tape volume onto another tape volume. Exactly one input volume (labeled or unlabeled) will be completely copied to the output tape, including copying all labels and data exactly as read from the input. However, image copy includes special functions, such as printing input tape labels (even though they are being copied as data) and copying empty data files (that would normally cause FATAR to stop prematurely).

If you want the tape management information for the files which you are copying to be updated to show that the files are on a new device, you should use the FATSCOPY IMAGE copy function. FATAR does not update the tape management records.

FATAR INPUT TAPE I/O

FATAR uses EXCP rather than any standard access method when reading input tape (from DD name TAPEIN) in order to have more control over error recovery processes and to bypass some access methods restrictions. FATAR can handle blocks up to 256K.

FATAR will issue a normal OPEN for the first file on each input volume, allowing security and tape management systems to perform their normal checking and authorization. However, if a tape volume contains more than one file, FATAR will switch files without any intervening CLOSE/OPEN to improve performance. If a data set occupies more than one input tape volume, FATAR will CLOSE at the end of the current volume and re-OPEN the next input volume using the current data set name. This allows FATAR to process any multi-file and/or multi-volume tape or tapes in one execution. See Section 21.6 for security considerations on multi-file tape volumes.

If FATAR is invoked as a sub-task of FATS, it will read the input tape from DD name TAPEx rather than TAPEIN, where "x" is the same as the digit on the FATS control statement ANALYZE(x). This is compatible with the FATS DD name.

INPUT TAPE DATA BLOCK PROCESSING

As FATAR begins processing each data file, it analyzes the DCB information (record format and logical record length) it has available. If labels are being processed, FATAR will extract this DCB information from them. However, if DCB parameters were supplied on the DD statement for the input tape, they will override the tape labels. This allows you to suppress FATAR deblocking operations (by specifying DCB=RECFM=U) or to correct invalid DCB information on the tape labels.

If the record format of the file is fixed (RECFM=F or FB) or variable (RECFM=V, VB, VS or VBS), FATAR will attempt to deblock each data block it reads from the input tape into its logical records. Note that FATAR does not depend on the blocksize from the labels or JCL but will read all the data in each tape block and then attempt to deblock it according to the format and logical record length found. In fact, if the records are variable length, the record length from the labels/JCL is also not used, but rather the length fields at the start of each record are used to locate the records. If the record format is variable spanned (VS or VBS), FATAR will deblock each record segment individually.

If FATAR finds an error during deblocking (such as a fixed length block that does not contain an exact number of full records, or a variable length block whose last record is too long for the length remaining in the block), FATAR will print an error message and treat that block as unblocked for printing. This may aid in identifying improperly formatted tapes, but remember that blocks that were read with data checks may cause these errors simply because the data is invalid.

When deblocking is in effect, FATAR will not print or make available for modification the block prefix (for ASCII Tapes) or the block descriptor word ("BDW" - for variable format blocks). However, each record (including the record descriptor word or "RDW" for variable format) is printed and available for modification.

The FATAR control statements SCAN and REPLACE (that validate and modify data respectively), will operate on individual records within the block, and can refer to specific records in specific blocks. The DROP statement can delete whole blocks or records within blocks. The PRINT statement always selects whole blocks but will print individual records within those blocks. The KEEP statement always operates on whole blocks.

If deblocking is suppressed or not possible, then the entire contents of each block (including any prefix or BDW) is printed or available for modification as if it were one single record.

ASCII TRANSLATION

If the input tape has ANSI labels, or if DCB=OPTCD=Q was specified on the TAPEIN DD statement, then all data records are translated from ASCII to EBCDIC as soon as they are read. By default, the ASCII-EBCDIC translation table used includes only the most common characters. If you need translation of other characters, such as the Euro character, see the ASCII= operand for custom translation tables.

Conversely, if the output tape has ANSI labels, of if DCB=OPTCD=Q was specified on the TAPEOUT DD statement, then all records are translated from EBCDIC to ASCII before they are written. By default, the EBCDIC-ASCII translation table used includes only the most common characters. If you need translation of other characters, such as the Euro character, see the ASCII= operand for custom translation tables.

However, if both the input and output tapes are in ASCII, the data will be copied without translation, so that no characters will be lost due to the translation tables.

The Euro character (€) and other special characters are not included in the default ASCII translation tables because the hex representation of these characters will vary depending on the national character sets in use. To support these characters, you must build custom translate tables that will properly translate those characters in your environment. See the description of the ASCII= operand for details on custom translate tables.

EXTENDED I/O ERROR RECOVERY

Because of the extensive error recovery built into the hardware of cartridge drives (often using recovery methods not available to FATAR), extended I/O error recovery is not used on cartridge tapes. Data checks on cartridge tapes are considered to be permanent.

21.3 MODIFICATION AND SCANNING

MODIFICATION AND SCAN STATEMENTS

FATAR Modification/Scan control statements are edited by FATAR and stored in a table at initialization. As each block is read by FATAR, this table is scanned to see if any of these statements apply to the current block.

Modification/Scan statements allow you to:

- Print the contents of any block.
- Drop a block or record, suppressing it from being written to the output tape.
- Keep a block, overriding FATAR's automatic drop of blocks that get data checks.
- Change the length of a block, making it shorter or longer than the original, and reposition the original data in the new block.
- Replace data at a given position in a block or record, optionally verifying the original contents first.
- Scan a block or record for any field containing a given value, and replace that with a new value (for either the first or every occurrence in the block or record).
- Validate that certain fields in a block or record contain valid packed or zoned (printable) decimal data.
- Validate that certain fields in a block or record contain valid data bytes according to a user-specified table, that can specify either all of the valid byte values or all invalid byte values.

More than one modification may apply to a given block or record, so FATAR will scan the table for all applicable control statements, executing them in the order that they originally appeared in the control statement input. Also, one statement may apply to more than one block or record; a range of blocks, or a range of records within a block may be specified, or a statement may apply to all blocks and records in a file (to provide a global Modification/Scan facility).

Normally, all blocks that are modified will be printed after modification, although this can be suppressed. If fields are being validated (the SCAN statement), FATAR will PRINT a warning about each field that fails validation. In any case, a message will be printed about every block that was affected by a control statement (documenting that it was "DROPPED", "PRINTED", etc.). If the control input contains Modification/Scan statements that are never selected by FATAR (if, for example, they specify a block number that does not appear on the tape), they will be identified at the end of the FATAR run so that you can investigate the reason.

Although the DROP, KEEP and REPLACE statements are most meaningful when an output tape is being produced, they will be processed normally even when no output tape is present. This allows you to "dry run" your control statements before you actually create an output.

RENAME STATEMENT

RENAME statements allow you to change the names of output files created by FATAR. They are processed like other Modification/Scan statements and stored in the same table. However, they are used only at the point that an output tape file is being opened. Normally, FATAR will use the data set name of the input file when opening the equivalent output file, but RENAME allows you to completely specify a new name (NEWN=), or specify a special string (NEWI=) that selectively modifies the input name, index level by level. Each RENAME statement can apply to one specific input file, or to all input files.

RENAME with NEWI= normally works only if the full data set name of each input data set is known (e.g., from tape management information), but the UNCOND operand on the RENAME statement will allow NEWI= to be used even when the full name is not known; the NEWI= will be applied to the truncated 17-character name from the tape labels (if the label name starts with a period, the period is discarded.

21.4 OUTPUT TAPE PROCESSING

OUTPUT TAPE PROCESSING

If the DD statement TAPEOUT is present, a logical duplicate of the input tape(s) will be generated. It is a "logical" duplicate since FATAR will allow the copy to have a different type of labels from the original, the tape density or tape type may be different, and user-specified modifications may be applied. The output tapes will normally have the same number of files as were found on the input tapes, in the same order, with the same data set names (though the RENAME statement and the output exit allow you to rename files on the output tape).

As each file is encountered on the input tape, the equivalent file will be opened on the output tape; if the input tape contains 5 files, 5 files will be created on the output, with the same contents as the input. However, it is possible to request FATAR to drop an entire file (e.g., DROP LF=2,B=ALL) in which case the output file corresponding to the dropped input file will not be opened at all.

OUTPUT LABEL PROCESSING

Label creation on TAPEOUT is totally controlled by the LABEL= operand on the TAPEOUT DD statement. The tape will be created as standard label (SL), ANSI label (AL) or unlabeled (NL or BLP).

FATAR will generate a data set name for each file opened on TAPEOUT; it may be recorded by your tape management system. If the output tape is labeled (LABEL=SL or AL), this name will also be recorded in the tape labels created. The name is derived by these rules:

- If the input tape is not labeled, or label processing is suppressed by LABELS=NO, the DSN= specified on the TAPEIN DD statement is used. However, if you have the tape management interface enabled (see Section 21.5) and the tape is recorded as unlabeled in the tape management database, the data set names recorded by tape management will be used.
- If the input tape is labeled and labels are read by FATAR, the data set name is extracted from the HDR1 label that precedes each data file. However, although data set names can be 44 characters, **only the last 17 characters** are recorded in the tape label. So FATAR does some additional processing to attempt to derive a proper data set name:
    - If the name read from the label matches the last 17 characters of the DSN= parameter provided on the TAPEIN DD statement, that name is used.
    - If you have the tape management interface enabled (see Section 21.5), the input tape is recorded as labeled in the tape management database, and the name from the HDR1 label matches the *end* of the full name of the equivalent input data set from that database, that full name is used.
    - If the EXIT= parameter was specified on the FATAR ANALYZE statement, a user-coded exit routine is called that can modify the output data set name (described later in this section).
    - If a RENAME statement (Section 23.3.7) was specified that applies to this input file, the replacement name (NEWN=) or name modification string (NEWI=) will be used to name the output data set.
- If none of the above provide the output name, the truncated 17 character name will be used; this may result in security errors.

If the output tape is labeled, the file sequence numbers of the output files created are controlled by the TAPEOUT DD statement. Normally, the first output file will be file sequence 1 regardless of the file sequence of the input file; the next output file will be 2, etc. If some input files are being dropped (via DROP LF=n,B=ALL) then no equivalent output file is created; there will never be gaps in the file sequence numbers created on the output tape. If you want to use FATAR to

add files to an existing tape, you can specify LABEL=n to specify the next available sequence number.

If labels are being processed on the input tape, DCB information read from the header labels of each file will be used to open the output file; if DCB= parameters are provided on the TAPEIN DD statement, they will override those from the tape labels. This DCB information may be recorded by your tape management system; if the output tape is labeled, DCB characteristics are recorded in the output tape labels.

***Note**:* If DCB parameters are coded on the TAPEOUT DD statement, they will be ignored. The DCB characteristics of each output file are always derived from the current input file's labels or from DCB overrides on the TAPEIN DD statement.

When output files are closed, an option (CAT= operand) can be used to cause FATAR to catalog the each file it created on the output tape. This allows you to easily do a logical copy of any input tape and recatalog the data sets to the new volume. This is especially useful when copying tapes to a new media type.

However, if FATAR was unable to identify the full data set name of the output data set using the rules above, cataloging will not be done. Since the truncated data set name from the tape labels is the *last* 17-characters of the name, it is not useful for cataloging. There are 2 exceptions:

- If NEWN= is used on a RENAME statement to provide a complete new name for a data set being copied, CAT= will be honored.
- If the name from the input tape labels is 16-characters or less (blanks at the end), it is the complete name so CAT= will be honored.

OUTPUT BLOCK PROCESSING

FATAR uses EXCP to write blocks to the output tape, so the output blocks will be an exact duplicate of the input blocks in each file regardless of DCB characteristics, unless reblocking or modification is requested.

By default, any block that gets a permanent data check when read from the input tape **will not** be written to the output tape, under the assumption that the data that was read is probably not usable.

After all modifications have been applied to a given block from the input tape, it will be written to the output tape (unless it is being dropped). Standard system error recovery is permitted on the output tape, so any WRITE error reported back to FATAR will cause the entire run to be terminated at that point.

If the output tape has ANSI labels (LABEL=(,AL)), or if DCB=OPTCD=Q is specified on the TAPEOUT DD statement), the block will be translated from EBCDIC to ASCII; for variable format tapes, the block format is converted to ANSI variable format.

Data set reblocking may be requested by the REBLOCK= keyword on the ANALYZE statement. If so, FATAR will reblock each copied file if it is fixed or variable format (except variable spanned and ANSI variable), if its logical record length is less than the new blocksize, and if its current blocksize is not greater than the value of MAXRBLK= (maximum blocksize to reblock) if specified. When reblocking, each logical record is copied into a reblocking buffer that is written to TAPEOUT only when full (or at end-of-file).

FATAR OUTPUT FILE EXIT

An optional user open exit may be used to modify the data set names of the output files. If specified by the operand EXIT=(OPENEXIT,exitname), the module "exitname" will be loaded and will be called before opening each output file copied to TAPEOUT. If "exitname" cannot be found FATAR will abend with a S806 upon parsing the EXIT= keyword statement.

The register conventions when the exit is called are (displacements are decimal numbers):

R1 -- Pointer to PLIST containing:

- +0 Pointer to a fullword containing a function code:

  4 is a normal entry, pointers at +8 to +16 are valid

  8 is last call: allows you to clean up exit environment (such as freeing work areas).
- +4 Pointer to a doubleword on a doubleword boundary that can be used by the exit routine for any purpose such as remembering the subpool, length, and address of any area the exit acquired for operation.
- +8 Pointer to TAPEIN JFCB
- +12 Pointer to TAPEOUT JFCB
- +16 Pointer to a 44-byte field containing the data set name that FATAR intends to use for TAPEOUT.

R13 -- Pointer to a standard save area

R14 -- Return address

R15 -- Entry point of exit routine

The registers should be saved and restored by the exit routine. Only the TAPEOUT JFCB should be modified; you can change the data set name and/or data set characteristics (RECFM, LRECL, etc.). The IBM macro IEFJFCBN can be used to define the JFCB fields.

Initially the data set name in the TAPEOUT JFCB and the data set name field pointed to by the PLIST+16 are the same; both are provided to simplify exit coding, using the data set name field as input while you modify the JFCB dsname. If the tape management interface is in use (see Section 21.5), the data set name in the TAPEIN JFCB will be that determined by the interface. If the RENAME statement was used, the output data set names will be the new name specified.

A sample exit routine (named "OPENEXIT") is included in the Installation Control Library (ICL) that was loaded from the FATS/FATAR installation tape (See Section 90). The sample exit replaces the high level qualifier (or everything in front of the first period) of the data set name with a new qualifier hard coded in the exit. This shell could be expanded to add additional functions. However, note that the RENAME statement (see Section 23.3.7) is an easier way of renaming FATAR output data sets in most cases.

MAKING AN IMAGE COPY

FATAR has a special mode where it will create an "image copy" of a tape. Image copy will essentially create an exact bit-for-bit copy of an input tape on an output tape, including an exact copy of all of the labels on the input tape, so the output tape will even have the volume serial of the input (although an option is available to preserve the output serial).

This function is useful when you need to replace a tape that is damaged by copying all the data to a new volume. If the input tape is in an Automated Tape Library, the new volume needs to be externally relabeled with the serial of the original tape and take its place in the tape library. Normally, no catalogs or tape management information need be updated since the new tape looks just like the original. (However, if you have the FATS/FATAR tape management interface enabled, you can force the copied files to be cataloged or recataloged by using CAT=RECAT.)

Image copy is invoked by specifying LABELS=IMAGE on the ANALYZE statement. LABELS=IMAGE is similar to the operation of the default LABELS=YES in that FATAR will

process and display all labels on the input tape and will automatically switch to LABELS=NO mode if the tape is not labeled. However, LABELS=IMAGE causes some differences from normal processing in order to accomplish the image copy:

- The input tape will always be rewound and processed from the beginning, even if you have OPENed it to another position with the LABEL= parameter in JCL.
- Only one input tape volume will be processed, even if you specify additional input volume serials.
- The OUTBLP option will be assumed, so that the output tape will be opened as an unlabeled tape. This is required so that FATAR can write the labels from the input tape onto the output.
- As labels and tape marks are read from the input tape, they will be written directly to the output; normally FATAR allows OPEN to create labels and tape marks on the output. FATAR will do only one OPEN on the output tape; it will not open every output file as it normally does.

The result is that the output tape will be a faithful, bit-for-bit image copy of the input tape volume (except that you can request that the output tape volume serial be preserved with the OUTSER=YES option). Of course, you can still use FATAR Modification statements to change data as it is being copied, but this is usually not done during an image copy.

Because the labels are copied as data, there are some restrictions when using image copy:

- Only one tape will be read in a FATAR step doing an image copy. If a multi-volume tape set must be copied, each tape must be copied in a separate step.
- For multi-volume tape sets, each tape volume must be completely copied to a separate output tape. Since all input volumes in a set (except the last) will contain data all the way to the end of the tape, you must be sure that the output tape used is large enough to contain all of the input tape data.
  - For virtual tapes, the capacity of each must be at least as large as the input tapes.
  - For physical tapes, that can come in various lengths, you must be sure that an output tape of the same length or longer is used. Even tapes that are supposed to be the same length may vary slightly, causing the copy to fail if the output is too short. Some tape vendors provide "extra length" cartridges just to solve this restriction. To try to avoid this problem, FATAR will attempt to write output blocks beyond the normal logical EOT (End-of-Tape) during an image copy; for cartridges, it will continue writing until the drive reports it cannot continue (Physical EOT); if the difference in the tape lengths is slight, this is usually enough to allow the copy to complete.
- Since FATAR does not OPEN each output file, they will not be recorded by your tape management system. For this reason, LABEL=EXPDT=98000 is usually coded on TAPEOUT (or whatever is required to bypass recording by your tape management system). Since image copy is usually used when a damaged or suspect tape is being copied to a new volume with the original serial number, this is not a problem since the new tape volume will simply replace the old volume in your tape library after being externally labeled with the input tape's volume serial; the existing tape management records are still valid for the new tape.
- If you want make an image copy which will copy tape management information for your data sets, you need to use the image copy function of FATSCOPY instead of using FATAR.

21.5 TAPE MANAGEMENT SYSTEM SUPPORT

FATAR contains support to interface to tape management systems to get complete information about the names of data sets being read by FATAR. This is often necessary because the tape labels contain only a truncated version of that name (the **last** 17 characters of the original name).

The supported tape management systems are:

- CA 1 (also called TMS) and CA TLMS, both from CA Technologies
- DFSMSrmm (also called RMM), IBM's tape management system

As part of the installation of FATS and FATAR, you must run the FATZAPOP program to tell FATAR which tape management system you are using. This installation procedure is described in Section 90.3.

If you have more than one tape management system in use, contact Innovation for assistance.

USING THE TAPE MANAGEMENT INTERFACE

Invocation of the tape management support in FATAR is automatic. FATAR will interface to your tape management database to query that volume serial.

The interface builds 2 tables used by FATAR:

1) A table of the full data set names of each file on the tape. If the tape is part of a multi-volume tape set, all data set names on the tape set are tabled.

2) If the tape is part of a multi-volume tape set, a table of all volume serials that are part of the tape set. The label type (SL, AL, NL) and the tape status (active, scratch) of each tape is also in this table.

Of course, the interface may report that the input volume serial was not found in the tape management database; this will be normal if the input volume is a foreign tape or if its true volume serial cannot be determined. For unlabeled tapes and tapes processed with LABELS=NO, FATAR will use the volume serial provided on the TAPEIN DD statement, that may not be the true serial. Other interface errors may prevent FATAR from getting the desired information.

USING THE TAPE MANAGEMENT TABLES

As FATAR reads each file from the input tape, it will attempt to match that file against the table of data set names, in order to get the full 44-character data set name.

For labeled input tapes it will extract the truncated 17-character data set name from the tape labels, and compare it against the end of the name recorded by tape management for this file. If they match, the full name is used by FATAR. If they do not match, the name from the labels is used. In either case, the name is reported in the FATAR listings.

For unlabeled input tapes, the name from the data set table will always be used.

If the input is a multi-volume tape set, as each new tape is mounted FATAR will verify that its volume serial is the expected next volume according to the tape management volume table. If it is not, a warning message is issued, but processing continues. An option (NEXTVOL=TMS) is available that causes FATAR to mount and process the volumes from the volume table, ignoring the JCL; this allows you to specify only the first volume in a multi-volume tape set and still have FATAR process the entire set.

21.6 SECURITY

PROTECTING FATAR EXECUTION AND OPTIONS

FATAR allows you to use your security system to control which users are authorized to execute FATAR and that are allowed to use various FATAR options. For example, you might authorize all users to analyze input tapes, but restrict the ability to copy tapes. **By default, the only resource that FATAR checks for is FATAR.DATASET.SECBYPAS. Security checking for the other resources is disabled by default; see Section 90 for instructions on enabling those security checks.**

Use of FATAR is controlled by resources defined in the RACF FACILITY class (or its equivalent in other security systems). Consult your IBM RACF or other security vendor documentation for the procedure for defining such FACILITY class resources. Once defined, you must grant READ authority to the resource for users authorized to use it.

For compatibility with earlier releases of FATAR, the default is to allow all users to execute FATAR and use all options. If your security system tells FATAR that a given resource name is not defined, all users will be able to use that function. If you define some but not all of the resources, only those defined functions are restricted. For example, if you don't define FATAR.COPY, all users can use FATAR to copy tapes.

However, some security systems, such as ACF2, protect all resources by default. If these security checks are enabled in such systems, no one will be able to use these FATAR functions until they are defined.

The FACILITY class resources that may be checked by FATAR are:

FATAR.ANALYZE ……..	execute FATAR; users not authorized will be unable to use FATAR at all
FATAR.COPY	provide a TAPEOUT DD statement (copy a tape)
FATAR.LABELS.YES.....	use the LABELS=YES or REQUIRED option (the default)
FATAR.LABELS.NO	use the LABELS=NO option
FATAR.LABELS.IMAGE	use the LABELS=IMAGE option to image copy a tape
FATAR.LABELS.ONLY	use the LABELS=ONLY option to print only labels on a tape
FATAR.LABELS.EOD	use the LABELS=EOD option to recover overwritten cartridge tapes
FATAR.PRINT.....	use the PRINT statement to print data
FATAR.RENAME	use the RENAME statement to rename copied data sets
FATAR.BLP.....	use the BLP operand or LABEL=(,BLP) in TAPEIN JCL
FATAR.OUTBLP	use the OUTBLP operand or LABEL=(,BLP) in TAPEOUT JCL
FATAR.EOD.....	recover data past a hardware EOD mark on cartridge tapes
FATAR.DATASET.SECBYPAS .	bypass data set security checking for input files

As each control statement is processed, FATAR will issue a RACROUTE to verify that the user has at least READ authority to the proper resource name. The step will fail if RACROUTE returns a code of 8 (or higher) meaning that the resource is defined to the security system but the user associated with the FATAR step does not have authority to it.

INPUT DATA SET SECURITY CHECKING

As FATAR opens each input tape, normal OPEN security checking is done, based on the parameters on your TAPEIN DD statement. However, if BLP is used, the data set name you provide may not be the real data set name on the tape; see notes below on the use of BLP.

For a multi-file input tape (TAPEIN), FATAR does one OPEN on the first file on each tape (or wherever the LABEL= operand on your TAPEIN DD points). For additional files, FATAR simply reads them, so no OPEN security check is done for those additional files. So, only the first file on each tape volume is security checked.

To close this security exposure, FATAR will issue a RACROUTE for the data set name in the DATASET security class, to verify that the user has at least READ authority to the data set name of each input data set. This check is done only if FATAR has the full data set name of the input data set; this is usually true only if you have installed the FATAR Tape Management Interface (Section 21.5) and the input tape is recorded by your TMS. Note that in some cases (such as LABELS=NO and LABELS=IMAGE), the data set names are not known and will not be checked.

Before issuing these checks, FATAR will check the FACILITY class resource FATAR.DATASET.SECBYPAS. If this resource is defined, and this user has at least READ authority to it, no DATASET checks will be done in this step. For compatibility with earlier releases of FATAR, if the resource is not defined to your security system, no DATASET checks will be done for any user.

If you don't define resource FATAR.DATASET.SECBYPAS, no additional input data set checks are done for any user; users may be able to access data with FATAR that they are not authorized to read. **If you do define it**, then data set checks are done for *all users* except those whom you authorize to that resource.

Note: **The input data set security checks are done by default; see Section 90.4 for instructions on disabling then.** In security systems that protect all resources by default, such as ACF2, FATAR.DATASET.SECBYPAS will appear to be protected, so data set security checks will be done for all users by default. If you don't want this, you must either disable the data set security check or define the FATAR.DATASET.SECBYPAS resource.

OUTPUT DATA SET SECURITY CHECKING

With the exception of image copy (LABELS=IMAGE), FATAR does a normal OPEN on every output data set on TAPEOUT, which will cause standard OPEN security checking to be done by the operating system. The user must be authorized to create a data set with the name specified by FATAR. LABELS=IMAGE does an OPEN only on the first output file.

When processing with LABELS=YES, the data set names extracted from the input tape will be used to open the equivalent output file. If you have the FATAR Tape Management Interface installed (Section 21.5), this is normally the full 44-character data set name.

If you do not have the TMS interface, the names may be the truncated data set names from the input tape labels (*last* 17 characters of the name); since there will be no applicable security rule for this name, no security check will be done.

USE OF BLP

When label processing on input or output volumes is bypassed using the BLP option in JCL or the BLP/OUTBLP operands on the ANALYZE statement, volume labels and data set names are not verified by OPEN and so authorization to the volume and data sets cannot be guaranteed; if anything, your security system will check authority to the volser and data set name provided in the JCL when BLP is used, these may not match the actual tape. For this reason, most security systems allow your installation to restrict the use of BLP.

However, some functions of FATAR (such as mapping or copying a tape with an unknown volser or dsname and doing an image copy) **require** the use of BLP. Innovation suggests that you provide BLP authority to a limited number of users (operators, tape librarians, and/or system programmers) so that they can run these functions when required.

Resources FATAR.BLP and FATAR.OUTBLP control the ability to use the BLP (for TAPEIN) and OUTBLP (for TAPEOUT) operands on FATAR control statements and the ability to put LABEL=(,BLP) on those DD statements. When opening a tape, FATAR will check if BLP is being used, from either source, and will fail the step if the resource is defined but the user is not authorized. If the user is authorized to FATAR.BLP or FATAR.OUTBLP but you have a general control on the use of BLP defined in your security system, the user must also be authorized to that resource.

READING DATA PAST EOD

FATAR.EOD controls the ability to read data past a hardware EOD (end-of-data) mark on a cartridge tape. The EOD mark was introduced with IBM 3490E drives and is used on all cartridge drives except 3480s. It is automatically written after the last file on the tape and normal access methods cannot read data past that mark; this is a security measure to prevent a user from writing a short file on a scratch tape and then accessing residual data past the EOD. In order to be able to recover data from accidentally overwritten tapes (see Section 24.2) FATAR can position past the EOD mark and read the residual data. If an EOD mark is encountered, FATAR will check for authority to FATAR.EOD before proceeding. You may want to protect FATAR.EOD to be sure that only authorized users can access data from overwritten tapes.

This page intentionally left blank.

22.0 FATAR EXECUTION JCL

Extensive ISPF panel support is available to generate job streams and control statements for FATAR; see Section 93.2 for further details.

To execute FATAR, the following JCL statements are used.

EXEC STATEMENT

The required EXEC statement specifies the FATAR program name, memory requirements (if your installation defaults are insufficient), and parameter field.

A region of 0M is recommended for FATAR; the program will use no more memory than is required for a particular function. .

The parameter field (PARM=) is used to define to FATAR the amount of memory required for storing FATAR modification and scan control statement information. FATAR requires about 30 bytes per control statement plus the data length of any D=, V=, or S= data fields given. The default table size of 32K bytes is sufficient for most FATAR applications, but if it proves to be inadequate, override it by `PARM='SIZE=nnnnnn'` where nnnnnn can be 1-999999.

Examples:

```
//FATAR   EXEC  PGM=FATAR,REGION=0M
//FATAR   EXEC  PGM=FATAR,PARM='SIZE=24576',REGION=0M
```

STEPLIB/JOBLIB DD STATEMENT

A STEPLIB or JOBLIB DD statement will be required if FATAR has been linkedited into a private library. It can be omitted if FATAR is in a system library that can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). This **must** be an APF authorized library.

SYSPRINT DD STATEMENT

SYSPRINT receives all FATAR printout and reports (except the tape summary), and is normally allocated to a SYSOUT data set. Its DCB attributes are RECFM=FBA,LRECL=121. If a blocksize is specified it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.

TAPESUMM DD STATEMENT

TAPESUMM is optional, but if present receives a summary of all files on each tape processed, including data set name, tape label information, block and byte counts, length of each file, minimum, maximum and average block sizes, and read error counts. Its DCB attributes are the same as for SYSPRINT.

SYSUDUMP DD STATEMENT

SYSUDUMP requests an abend dump if major errors occur; note that most internal abends in FATAR are for the user's information only and do not cause dumps. SYSUDUMP is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by Innovation. If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced:

```
//ABNLIGNR  DD  DUMMY
```

TAPEIN DD STATEMENT

TAPEIN specifies the tape or tapes to be analyzed. If more than one volume serial is specified, the volumes should be part of a single multi-volume tape set, in the proper order. If the input data set is cataloged, UNIT= and VOL= can be omitted, but be aware that for multi-volume, multi-file tapes, each catalog entry points only to the volumes that file occupies, so you may have to specify volume serials in the JCL in order to process the entire tape set. However, if you have the tape management interface enabled (see Section 21.5), you can code the NEXTVOL=TMS option that will use tape management information to mount and process all the volumes in the tape set automatically.

LABEL= can specify SL (the default), AL, NL, or BLP; FATAR's decision to process the tape as labeled or unlabeled is independent of JCL. However, the LABEL= parameter can be used to position the input tape to a particular file at which to start processing. If SL or AL is specified, the data set name of the first file to be processed must be properly specified, and FATAR will position to the header labels of that input file before starting. Since this is an input file, DISP=OLD is appropriate.

If LABEL=(,BLP) is permitted in your installation, its use on TAPEIN is desirable to be able to analyze unknown tapes, or to analyze tapes without having to update the JCL with the proper volume serial and data set name. Even if BLP cannot be specified in JCL, the BLP operand on the FATAR ANALYZE statement can be used to invoke BLP processing internally. See Section 21.6 for information on the security considerations for BLP.

If labels are being processed from TAPEIN, FATAR will usually receive sufficient information from those labels to deblock the data blocks in each file. However, if label information is missing or incorrect, it can be supplied by the DCB= parameter on TAPEIN. Even if label information is present, FATAR can be forced to treat input blocks as single large records for printing and modification by putting DCB=RECFM=U on the TAPEIN statement (the DCB information in the labels will still be used to open TAPEOUT files).

If LABEL=(,AL) or DCB=OPTCD=Q appears on the DD statement, or if FATAR detects ANSI labels on the tape, all labels and data read will be translated from ASCII to EBCDIC.

For image copy (LABELS=IMAGE), TAPEIN should specify only one volume serial; additional serials are ignored.

Examples:

```
//TAPEIN  DD   DSN=FILE.NAME,UNIT=TAPE,
//             VOL=SER=111111,DISP=OLD
//TAPEIN  DD   DSN=ANY.FILE,UNIT=TAPE,
//             VOL=SER=(222222,333333),
//             LABEL=(4,BLP),DISP=OLD
//TAPEIN  DD   DSN=CATALOG.ED.DATASET,DISP=OLD
```

TAPEOUT DD STATEMENT

The TAPEOUT DD statement is optional. If TAPEOUT is omitted, FATAR will simply analyze the input tape, but if present specifies a tape on which to create a copy of the input files read from TAPEIN. The LABEL= parameter on the TAPEOUT DD statement may be anything (SL, AL, NL, BLP) and FATAR will create the specified type of labels. Note that SUL or AUL is not required for the creation of user labels on TAPEOUT; if the input tape contains user labels, they will be copied to TAPEOUT (if labeled) unless the SUPULAB parameter is specified. Since this is an output file, DISP=NEW is usually appropriate. Volume serials may be specified, or may be omitted to call for scratch tapes.

If labels are not being processed on TAPEIN, the DSN= JCL parameter on the TAPEOUT DD is ignored and the DSN= specified on TAPEIN will be used to name every file created on TAPEOUT. If labels are processed on TAPEIN, the DSN= is still ignored, and every file name

will be copied from TAPEIN labels. The RENAME statement can be used to override the output data set names in both cases. If LABELS=OS is specified, the DSN= on the TAPEOUT DD **will** be used to name the output data set.

Also, if LABELS=EOD is specified to recover data from an overwritten tape, the TAPEOUT DD statement must include a DSN= specifying the output data set name for the first (partially overwritten) data set to be recovered; additional recovered files will be named according to the names in their header labels or RENAME statements.

The LABEL subparameters EXPDT= or RETPD= may be specified to provide an expiration date for the output files created. If omitted, the expiration date of the data set from your tape management system or the expiration date of labeled files read from TAPEIN will be used.

If LABEL=(,AL) or DCB=OPTCD=Q appears on the DD statement, all data written will be translated from EBCDIC to ASCII.

The ability to provide a TAPEOUT DD statement (i.e., the ability to copy tapes with FATAR) may be restricted by security rules; see Section 21.6.

***Note**:* If DCB parameters are coded on the TAPEOUT DD statement, they will be ignored. The DCB characteristics of each output file are always derived from the current input file's labels or from DCB overrides on the TAPEIN DD statement.

Examples:

```
//TAPEOUT DD DSN=NOT.USED,UNIT=TAPE,DISP=(,KEEP)
//TAPEOUT DD DSN=FILE.NAME,UNIT=CART,
//           VOL=SER=444444,DISP=(,KEEP)
```

SYSIN DD STATEMENT

The SYSIN DD statement is the source of FATAR control statements. It is normally a "DD *" spool file, but can be any disk or tape file with DCB characteristics RECFM=FB and LRECL=80. It can also be omitted entirely, in which case FATAR will assume all defaults and no modification or scan statements.

JCL WHEN FATAR IS A FATS SUB-TASK

FATAR may be invoked as a sub-task of its companion product FATS in order to execute more than one FATAR task simultaneously in the same jobstep. In this case several differences from the above rules apply (in each case, the "x" referred to is the number "x" in the FATS control statement ANALYZE(x)):

- FATAR will write its report to Print DD statement SYSPRINx rather than SYSPRINT (e.g., SYSPRIN1).
- The TAPEx DD statement rather than TAPEIN will be used for the input tape (e.g., TAPE1).
- If present, the TAPExOUT DD statement rather than TAPEOUT will be used for the output tape copy.
- Most ANALYZE statement parameters can be specified in the FATS ANALYZE(x) control statements. However, if you wish to override what FATS has specified, or wish to specify scan or modification control statements, use a SYSINx rather than a SYSIN DD statement (e.g., SYSIN1).

This page intentionally left blank.

23.0 FATAR CONTROL STATEMENTS

23.1 GENERAL

GENERAL RULES

All FATAR control statements must be input on 80-character records, of which only columns 1 to 71 are used by FATAR.

Every FATAR control statement must begin with an operation keyword from the list below. This keyword may begin in column 1 or be preceded by any number of spaces. It must be followed by one or more spaces. Every operation keyword accepts a number of operands that follow those blanks and are separated by commas with no intervening spaces. Control statements may be continued onto multiple input records by following the last operand with a comma and a space, and starting the next operand on the next record in any column (preceded by spaces).

Any FATAR control statement may contain comments; FATAR stops its scan of the control statement whenever it finds an operand followed by a space (or comma-space for continuation), so the rest of the statement may be filled with any comment desired. Any control statement record containing an asterisk ("*") in column 1 will be bypassed by FATAR and may entirely contain comments.

The FATAR operation keywords are:

ANALYZE.....Perform FATAR analyze/copy

COPY .....Same as ANALYZE except that TAPEOUT is required

DROP.....Do not copy one or more blocks

KEEP.....Copy a block even if permanent data checks or change its length

PRINT .....Print one or more blocks

REPLACE.....Replace data in a block or record

SCAN .....Validate data fields in a block or record for packed or zoned numeric or against a user-defined table

TABLE.....Define valid characters for a scan operation

RENAME.....Rename one or more files while copying

23.2 ANALYZE STATEMENT

ANALYZE/
COPY

ASCII=tabname
,BLP
,BYTEFACTOR=1000|1024
,CAT=NO|YES|RECAT|ONLY
,CHAR|HEX|DUMP
,CHARTAB=tabname
,CHECKTMS
,COMBFILES
,COMPACTION=SYSTEM|COPY|COMP|NOCOMP
,CRDATE=TODAY|COPY
,DUMP=NO|YES|ASK
,ENDAFTER=(LF=nnnnn**,B=**bbbbbbbbbb**)**
,EODRETRY=nnn
,EOTBLOCKS=nnn
,EOV=NO
,EXIT=(OPENEXIT,exitname**)**
,EXPDAYS=nnn
,KEEP
,LABELS=YES|NO|OS|REQUIRED|IMAGE|ONLY|EOD
,LBLPRT=FORMAT|CHAR|HEX|DUMP|NONE
,LENCHK
,LINECNT=nnnnn
,MAJERR=nnnnn
,MAXERR=nnnnn
,MAXTERR=nnnnn
,NONMETRIC|METRIC
,NEXTVOL=JCL|TMS
,MSGLEVEL=0
,NUMFILES|NF=nnnnn
,OUTBLP
,OUTDSN=YES
,OUTFSEQ=nnnnn
,OUTSER=YES
,OUTVOL=INVOL
,PRINTTMS=YES
,PRTLEN=nnnnnn
,REBLOCK=nnnnnn**,**
MAXRBLK=mmmmmm
,RECUSERL
,RETCODE
,RETRY=nnn
,ROR=NO
,SETNZRC
,STOP
,SUPULAB
,VALIDATE=INPUT|OUTPUT|ALL|INEXP
,VERIFY
,VIRTUNIT=esoteric
,WTO

An ANALYZE or COPY control statement is required (unless FATAR control statement input is omitted altogether). However, all of its operands are optional. Operands may appear in any order on the statement. Operands are available to control the input tape operation, error recovery actions and printing plus a few others; they are presented in those categories below. One ANALYZE/COPY statement must be first in the input.

The COPY statement is the same as ANALYZE, except that it requires the presence of a TAPEOUT DD statement and always copies the input tape.

If ANALYZE/COPY has no operands, or if the control statement data set is omitted altogether, FATAR will perform the following functions:

- Input tape will be tested for presence of labels. If not labeled, all files on one tape will be read up to a double tape mark (end of tape). If labeled, all files on all input volumes will be read until a double tape mark is read or the list of volumes is exhausted (EOV labels will cause tape switching).
- If labels are ANSI, all labels and data will be translated from ASCII to EBCDIC.
- If present, all labels will be printed. If the TAPESUMM DD statement is present, a summary of the files read will be printed for each input tape.
- All temporary and permanent data checks will be noted, including printing 80 bytes from the blocks preceding and following the data check block.
- If DCB information is available from tape labels or JCL, all blocks on TAPEIN will be checked to ensure that they don't exceed the blocksize; that they contain validly formatted fixed or variable records; and, for variable spanned records, that the record segments are valid.
- If the TAPEOUT DD statement is present, all data read from TAPEIN will be written to TAPEOUT, except that data check blocks will be dropped. The label type of TAPEOUT is controlled by the JCL for TAPEOUT. If TAPEIN is labeled, the output data sets will have the same names as the input data sets.

23.2.1 INPUT TAPE OPERANDS

ASCII= Specifies the name of a load module containing a pair of 256-byte translate table, one for translation of ASCII data to EBCDIC, the other for translating EBCDIC to ASCII. This will be used by FATAR whenever ASCII translation of input or output data is required. It can be used to include special characters, such as the Euro character, in the translation. The load module will be LOADed by FATAR so it must be in the FATAR load library or some other library accessible to FATAR. See Section 90.4 for instructions on creating your own translate table.

Default: An IBM translate table is invoked via the IBM XLATE macro. This IBM-provided table includes only common ASCII and EBCDIC characters and excludes most special characters (such as the Euro).

BLP The BLP operand is provided to help when installation conventions do not allow LABEL=(,BLP) to be specified on a JCL DD statement, but it is desired to bypass label processing on TAPEIN so that FATAR can open and process any tape (such as when the true volume serial is unknown). When BLP is specified on the ANALYZE/COPY statement, the JCL label parameter will be changed to BLP internally before TAPEIN is opened. If a file number is specified in the JCL it is not changed; for example, LABEL=3 will become LABEL=(3,BLP). See Section 21.6 for security considerations for the use of BLP.

CHECKTMS Issue tape management queries against input files (for information such as the data set name, expiration date, etc.) even though EXPDT=98000 is specified on the TAPEIN DD.

Default: Bypass tape management queries when EXPDT=98000 is coded on TAPEIN.

ENDAFTER= The ENDAFTER operand causes FATAR to stop reading TAPEIN after a certain block in a given file has been processed; it is much like the NUMFILES= parameter above except that it can terminate at any point within a file. "nnnnnn" is the logical file number (see "Modification/Scan Operands" in Section 23.4) and "bbbbbb" is a decimal number giving the block number (the first block in each file is block 1).

"n" may be 1-196605, "b" may be 1-2147483647

EOV=NO If LABELS=NO has been specified to suppress processing of TAPEIN tape labels but the tape is actually labeled (such as when doing an "image copy"), FATAR will still perform 2 label checks: it will identify EOV labels since they are **not** followed by two tape marks, and FATAR cannot otherwise tell when it has reached the end of the tape. Also, if copying to an output tape, block counts will be updated in any EOF or EOV labels encountered. Specifying EOV=NO suppresses both of these actions.

LABELS= Controls processing of tape labels on TAPEIN. Use of certain values for LABELS= may be restricted in your installation by security rules; see Section 21.6. The examples in Section 26.2 will help you understand when to use each option.

YES specifies that the input tape is to be checked for standard IBM labels or ISO/ANSI labels. If present, these labels will be processed and printed, and the data set name and DCB information will be used for copying the data set. If not present, the tape will be treated as unlabeled (as if LABELS=NO was given).

NO forces the input tape to be treated as unlabeled, even if it does have labels; all physical files will be treated as data. This can be used on a labeled tape to perform an "image copy"; however, LABELS=IMAGE is the preferred method for performing image copies.

OS requests that the operating system is to handle all label processing instead of FATAR. In this mode FATAR does an ANALYZE or COPY of a single file only. NUMFILES=1 is assumed and JCL parameters control whether TAPEIN is treated as labeled or not.

REQUIRED is the same as LABELS=YES, but in addition, if VALIDATE=ALL or INPUT is specified, the input tape will be rejected if it is not labeled.

IMAGE invokes a special "image copy" mode, used to make an exact bit-for-bit duplicate of a single input tape volume on an output tape. It acts similar to LABELS=YES for processing of labeled and unlabeled input tapes. See Section 21.4 for details.

ONLY is similar to LABELS=YES except that no output tape is allowed, and data files are rapidly skipped over. It provides a way to quickly print information about all labeled files on the tape without having to analyze the associated data files.

EOD is a special copy mode for recovering data from overwritten tapes. All cartridge drives except 3480 write a EOD (End-of-Data) mark after the last block written to the tape; the residual data on an overwritten tape is beyond the EOD. LABELS=EOD causes FATAR to search for the first EOD mark on the tape and recover the data beyond it. If the first data it can read past the EOD is a header label, it will process in LABELS=YES mode. If the first data is a data record, it will process that partial file in LABELS=NO mode, but if there are additional labeled files on the tape past the overwritten file, it will process them in LABELS=YES mode. Note that FATAR may not be able to determine if the data beyond the EOD was compressed; if you want the output tape to be compressed you should specify COMPACTION=COMP.

Default: LABELS=YES

***NOTE**:* LABELS=YES will be automatically changed to LABELS=NO if the first block read is not a label; this means that LABELS= rarely needs to be used since FATAR will properly handle a labeled or unlabeled tape automatically.

NEXTVOL= Controls how FATAR determines the next input volume for multi-volume input tape sets.

JCL causes FATAR to mount and read all of the volumes specified by the TAPEIN DD statement. These volumes must all be part of a multi-volume tape set, in the proper order. Note that if the tape set contains multiple data sets (multi-volume multi-file) and the TAPEIN DD references a data set on the tape via the system catalog (UNIT and VOLUME not given), it may not mount all the tapes in the set; in this case you will usually have to specify all the volumes in the set in the JCL.

TMS causes FATAR to use the table of tape serials obtained from your tape management system to mount and read all the tapes in the multi-volume set. The first tape is determined by the TAPEIN DD statement, but any other volumes that follow that first volume are taken from the tape management information. If the FATS/FATAR tape management interface has not been installed (See Section 90.3) or if the first input tape is not recorded in your tape management database, it will revert to NEXTVOL=JCL processing.

Default: NEXTVOL=JCL

NUMFILES=
NF= "n" is a decimal number specifying the number of logical files that FATAR is to read from TAPEIN. A logical file is one standard label file if labels are being processed, or one physical file if not. If labels are processed and an input file extends to more than one volume, it is still considered one logical file. If "n" is 0, FATAR will read all files until two consecutive tape marks are read (signifying the logical end-of-tape), or until all input tape volumes have been processed. Also, if labels are not being processed and "n" is 0, FATAR will stop if EOV labels are read. Note that if "n" is a value greater than the actual number of files on a tape, round tapes may run off of the reel and require operator intervention to rethread; this cannot occur on cartridge tapes. "n" may be 1-196605.

Default: NUMFILES=0

PRINTTMS=YES If the FATS/FATAR tape management interface is installed, FATAR will query your tape management system using the volume serial of the first input tape; the interface will return a table showing all data sets on the tape (or tape set) and all volumes in the tape set (See Section 21.5).

RECUSERL By default, FATAR recovers tapes as if they contain standard labels when LABELS=EOD is specified. If the overwritten tape contains user labels and RECUSERL is specified, FATAR will recover all of the overwritten files on the tape, except for the first file, as if they contain user labels. Call Innovation for assistance if you need assistance recovering an overwritten tape.

STOP When specified, message FATSW12 will be written to the operator at the end of each file read from TAPEIN, giving the option of terminating FATAR or continuing. This provides a measure of dynamic control of FATAR.

VIRTUNIT= Specifies the esoteric name (1-8 characters) assigned to virtual tape drives. Tape data sets cataloged to the device type associated with this esoteric name will be identified as VIRTUAL in reports.

WTO When specified, message FATSW11 will be written to the operator at the end of each file read from TAPEIN or verified from TAPEOUT, giving the block and error counts.

23.2.2 ERROR RECOVERY OPERANDS

EODRETRY= "n" is a decimal number specifying the number attempts that FATAR will make to reposition and continue after an EOD (End of Data) mark is encountered on cartridge tapes. 3490E, 3590 (Magstar), and similar drives record an EOD mark after the last valid data on an output cartridge; EOD marks are usually encountered by FATAR only when trying to recover data from an overwritten tape; the EOD mark precedes the data to be recovered. Note that 3480 drives do not record an EOD mark. Depending on the type and vendor of the drive, FATAR will make the specified number of attempts to position past the EOD and read the overwritten data; it may not be successful, and even if successful may be forced to space past some of the original data. "n" may be from 0 to 255.

Default: EODRETRY=20

KEEP By default, FATAR will not copy to TAPEOUT any blocks that had permanent data checks when read from TAPEIN. If KEEP is specified, FATAR will write to TAPEOUT whatever data it was able to read from TAPEIN during the last recovery attempt on a data check block. Note that the block length may be incorrect due to the data check.

MAJERR= If "n" major errors (FATS046/7/8/9 messages) are detected on TAPEIN, FATAR will issue message FATS016 and abend with a U0888 Abend. "n" may be 1-32767.

Default: MAJERR=1

MAXERR= If "n" permanent data checks are detected on TAPEIN, FATAR will abend with a U0888 Abend. "n" may be 1-32767.

Default: MAXERR=20

MAXTERR= If "n" temporary data checks are detected on TAPEIN, FATAR will abend with a U0888 Abend. A temporary data check is one that FATAR reads successfully during recovery processing. "n" may be 1-32767.

Default: MAXTERR=100

RETRY= "n" specifies the number of times FATAR will attempt to re-read a block on TAPEIN that results in a data check before considering the error permanent. FATAR will attempt this many retries in each direction (forward read and backward read). "n" must be from 0 to 256. "RETRY=0" suppresses all re-read attempts and immediately considers the error to be permanent; if a tape drive cannot seem to "get past" an error on a tape, this may allow the tape to be read. RETRY= is used only on round (3420-type) tapes; for tape cartridge systems, because of the extensive error recovery done by the tape control unit, RETRY=0 will be forced and cannot be overridden.

Default: RETRY=40 on round tapes, RETRY=0 on cartridges

ROR=NO ROR=NO suppresses "READ-OPPOSITE-RECOVERY", the attempt to recover from a data check by reading it in the backwards direction. With this parameter specified, FATAR will only attempt forward re-reads for data checks. ROR is used only on round (3420-type) tapes.

23.2.3 OUTPUT TAPE OPERANDS

COMPACTION= If TAPEOUT is a cartridge tape capable of writing compacted (IDRC) format, COMPACTION= (which can be abbreviated COMP=) controls compaction on the output tape.

SYSTEM uses the system default for your installation.

COPY will use compaction if the input file was compacted; otherwise the output file is not compacted.

COMP will always compact the output tape.

NOCOMP will never compact the output tape.

Following z/OS standards, labels on cartridge tapes are never compacted by FATAR.

Default: COMPACTION=COPY. However, if LABELS=OS is specified, compaction is controlled by the TAPEOUT DD statement (if the TRTCH=COMP or NOCOMP operand is not specified on the DD statement, the system default will be used).

Note: If you are recovering data beyond end-of-data with LABELS=EOD, FATAR may not be able to determine if the recovered data was compressed, so you may need to specify COMPACTION=COMP if you want the output tape to be compressed.

CAT= Controls cataloging of tape files copied to the output tape:

NO - output tape files will not be cataloged.

YES - output tape files will be cataloged if they are not already cataloged.

RECAT - output tapes files will be cataloged; if they are already cataloged, the catalog will be updated.

ONLY – output tape files will be cataloged only if the equivalent input tape files are cataloged (the catalog must point to the matching volser and file sequence number of the input file).

If you are copying a tape and are not renaming the output files (no RENAME statement), CAT=RECAT or CAT=ONLY is required to change the catalog to point to the new output volume and device type.

Cataloging will be attempted only if FATAR knows the full output data set name; in most cases this is true only if the FATS/FATAR tape management interface is enabled; see Sections 21.4 and 21.5 for details.

Cataloging can be done for an image copy (LABELS=IMAGE) if CAT=RECAT is used. In this case, LBLPRT=FORMAT is forced.

Default: CAT=NO

CRDATE= If TAPEOUT is a labeled tape, the creation date recorded in the labels on TAPEOUT will be the current run date, unless CRDATE=COPY is specified, causing the tape label creation date to be copied from the corresponding labels on TAPEIN if it is also labeled. CRDATE=COPY is not recommended for CA 1 (TMS) tape management system users, since (a) it affects the expiration date on TAPEOUT in unpredictable ways and (b) the creation date stored in the TMC will always be the current run date.

Default: CRDATE=TODAY

EOTBLOCKS= When performing an image copy (LABELS=IMAGE), the output tape volume must always be long enough to hold all of the data on the input tape volume. However, when the input volume was originally written all the way to the end (usually because it was part of a multi-volume tape set), minor differences in tape lengths may make the chosen output tape just slightly too short to hold all the input data before logical EOT (End of Tape) is reached. EOTBLOCKS= specifies the number of additional blocks FATAR will attempt to write to TAPEOUT even after reaching logical EOT. "n" may be a value from 1-999.

Default: EOTBLOCKS=999 on cartridge tapes.

OUTBLP OUTBLP is provided to help when installation conventions do not allow LABEL=(,BLP) to be specified on JCL DD statement, but it is desired to bypass label processing on TAPEOUT so that FATAR can open and process any tape (such as when the true volume serial is to be overwritten in an image copy). When OUTBLP is specified on the ANALYZE/COPY statement and DEFER is coded in the UNIT parameter of the DD statement, the JCL LABEL parameter will be changed to BLP internally before the tape is opened. If a file number is specified in JCL it is not changed, e.g., LABEL=3 will become LABEL=(3,BLP). See Section 21.6 for security considerations for the use of BLP.

OUTBLP is forced if LABELS=IMAGE is specified, since BLP processing on the output tape is required for image copies (See Section 21.4).

MAXRBLK= "n" is a decimal number (4-262144) giving the largest input blocksize to be considered for reblocking. MAXRBLK= is meaningful only if REBLOCK= is also specified. Any input file whose blocksize on TAPEIN is greater than "n" will not be reblocked.

Default: All files are reblocked if REBLOCK= is specified.

OUTDSN=YES When LABELS=IMAGE is specified (an "image copy"), and VALIDATE=OUTPUT or ALL is specified, OUTDSN=YES requests FATAR to verify that the data set name in the labels of the output tape consists of 17 character zeros ("0"). This is the state that a tape freshly initialized by FATS or by IEHINITT will be in, so this verifies that the output tape is a newly initialized tape.

Default: The output data set name is not verified.

OUTFSEQ= Specifies the file sequence number (1-65535) to be assigned to the first file written to TAPEOUT. This should be used only when adding files to an existing output tape; specify a value one larger than the last file on the tape. Note that values up to 9999 can also be specified by the LABEL= operand on the TAPEOUT DD statement.

Default: OUTFSEQ=1

OUTSER=YES When LABELS=IMAGE is specified (an "image copy"), OUTSER=YES will cause the volume serial of the output tape mounted to be preserved, i.e., the image copy is modified so that the output tape volume serial is inserted in the volume label (VOL1) and data set labels as the data is copied. In most cases, this type of copy should be done as a logical copy (LABELS=YES), but OUTSER=YES can be used when the exact layout of the tape must be preserved.

Default: The volume serial of the input tape will be copied to the output during an image copy.

OUTVOL=INVOL When LABELS=IMAGE is specified (an "image copy"), and VALIDATE=OUTPUT or ALL is specified, OUTVOL=INVOL requests FATAR to verify that the tape mounted on TAPEOUT has the same volume serial as the tape on TAPEIN, in other words, that the output tape has been initialized and labeled with the same volser as the input tape in preparation for the image copy.

Default: The output tape serial will be used for validation (if requested).

REBLOCK= "n" is a decimal number (4-262144) specifying the target blocksize for data reblocking. When this keyword is specified, all files will be reblocked as they are copied to have blocksizes as close to "n" as possible. MAXRBLK= may be specified to bypass reblocking on files that already have an acceptably large blocksize.

Note: Whenever you reblock tapes, you must be sure that the applications that will read the tapes are capable of handling the new blocksize.

Default: Blocks are copied exactly as read from TAPEIN, without reblocking.

SUPULAB If any file read from the input tape contains user labels and the output tape is a labeled tape, the user labels will be copied to the output tape unless SUPULAB (“suppress user labels”) is specified.

This may be needed when you have an encryption product that writes its own user labels that contain the file’s encryption keys.

Default: User labels copied if output tape labeled.

23.2.4 PRINT CONTROL OPERANDS

BYTEFACTOR= Whenever byte counts are rounded to kilobytes or megabytes, BYTEFACTOR specifies the divisor used to calculate the values. Only 2 values are accepted: BYTEFACTOR=1000 (the default) divides the byte counts by 1000 or 1000000. BYTEFACTOR=1024 divides by 1024 or 1048576.

Default: BYTEFACTOR=1000

CHAR
HEX
DUMP

CHAR specifies that FATAR will print any required blocks or records in EBCDIC, 80 characters per print line.

HEX specifies that FATAR will print in hexadecimal format, 8 groups of 4 bytes (32 bytes total) per print line.

DUMP specifies a combination, with the 32 characters of EBCDIC printed to the right of the same 32 bytes of data in hexadecimal.

This format can be overridden for specific blocks on a PRINT control statement.

Default: CHAR

CHARTAB= Specifies the name of a load module containing a 256-byte translate table. This will be used by FATAR whenever it translates hexadecimal data to EBCDIC for printing. This can be used for special printer character sets or for special national characters. The load module will be LOADed by FATAR so it must be in the FATAR load library or some other library accessible to FATAR. See Section 90.4 for instructions on creating your own translate table. The translate table name can be anything you like, except that FATCHRTB cannot be used.

Default: An internal translate table (FATCHRTB) is used that prints standard US printer numerics, uppercase alphabetics and normal special characters. All other characters are translated to periods.

LBLPRT= Whenever FATAR recognizes labels on the input tape, it will normally print those labels (even if label processing has been suppressed by LABELS=NO). Formatting of labels is independent of the CHAR, HEX, and DUMP operands described above. LBLPRT= specifies an overriding print format for labels only. The values CHAR, HEX, and DUMP have the same meanings as the equivalent operands described above. **LBLPRT=FORMAT** causes the labels to print in character format, followed by a two line breakdown of the individual fields in each label so that the contents of those fields can be determined more easily. **LBLPRT=NONE** suppresses label printing. Labels will be recognized and printed even when label processing has been suppressed by LABELS=NO.

Default: LBLPRT=FORMAT

LINECNT= "n" specifies the number of lines per page (1-32767) to print on all FATAR print data sets.

Default: LINECNT=60

MSGLEVEL=0 By default, FATAR will print the contents of any block that has been modified by a REPLACE or KEEP statement (up to the limit specified by PRTLEN=). MSGLEVEL=0 will suppress this print (FATAR will continue to print a line documenting that the block was modified).

PRTLEN= Whenever a block from the input tape is to be printed, for whatever reason, FATAR will print only the first "n" bytes or characters of data (0-262144) unless overridden by a PRINT control statement. If the block is being deblocked, FATAR will print as many logical records as exist in the first "n" bytes, and may truncate the last record.

Default: PRTLEN=80

NONMETRIC
METRIC When NONMETRIC is specified (or defaulted) FATAR will list tape lengths and error positions and lengths in U.S. units (feet and inches). When METRIC is specified FATAR will list tape lengths and error positions and lengths in metric units (meters and centimeters).

Default: NONMETRIC

23.2.5 MISCELLANEOUS OPERANDS

COMBFILES Combine all files found on TAPEIN into a single file on TAPEOUT. The tape labels on TAPEOUT will show the DCB characteristics copied from the first file on TAPEIN; COMBFILES is intended for use when all the files on TAPEIN have similar characteristics. If the DCB characteristics of the TAPEIN files vary, various FATAR errors may occur, and any application that reads the combined output tape may also experience errors.

DUMP= Controls whether an ABEND dump will be produced for errors that don't usually produce a dump, such as U0100, U0888, U0913, and U0502 ABENDS.

YES – produce a dump for these ABENDs (if a SYSUDUMP or SYSABEND DD is provided)

NO – do not produce a dump

ASK – ask the operator, via a WTOR, if a dump should be produced.

Default: DUMP=NO

EXIT= Activate the FATAR user open exit facility described in Section 21.4, "Output Tape Processing". The parameter "exitname" is the name of the load module to be loaded by FATAR and executed at open time for TAPEOUT.

Default: No exit is activated.

EXPDAYS= Used with VALIDATE=INEXP, documented below.

Default: EXPDAYS=0

LENCHK If specified, FATAR will compare the length of every block read to the length of the preceding block and print the contents of the block if the lengths do not match. The first block in every file will always print. This can be used to scan a fixed length file for improper length blocks.

RETCODE RETCODE causes FATAR to terminate with a return code (condition code) of 12 rather than a U0200 or U0888 abend when serious errors are found.

SETNZRC SETNZRC causes FATAR to terminate with a nonzero return (condition) code when various unusual conditions occur, with a value of 4 or 8, instead of zero. See Section 40.1 for details.

VALIDATE= If LABEL=(,BLP) is specified on TAPEIN or TAPEOUT DD Statements, data management will not verify the volume serial or expiration date of the tapes. However, if VALIDATE= is specified, FATAR will verify the volume serial of TAPEIN (VALIDATE=INPUT), TAPEOUT (VALIDATE=OUTPUT) or both (VALIDATE=ALL). The volume serial compared is the serial in the DD statement. For TAPEOUT, the expiration date in the tape label will be checked. If the wrong volume is mounted or if the expiration date is not yet reached, FATAR will issue message FATSW08 to the system operator giving the option of ignoring the error, mounting another tape, or terminating. TAPEOUT validation applies only to the first output volume.

In addition, VALIDATE=INEXP will validate the input volume serial and will also validate the expiration date of the input tape. If the input tape is expired, message FATSW08 will be issued allowing the operator to bypass copying tapes that have expired. If the EXPDAYS=nn operand is also specified, FATSW08 will be issued for expired input tapes and those that will expire in the next "nn" days, allowing you to bypass copying tapes that will expire shortly.

Default: No validation

VERIFY If VERIFY is specified and an output tape is being produced, at the conclusion of the copy FATAR will reopen the output tape as input and re-read it to verify its readability and contents. (It is **not** compared to the input tape.) All FATAR options are reset to their defaults and all Modification and Scan control statements are ignored during the verification.

23.3 MODIFICATION/SCAN STATEMENTS

MODIFICATION & SCAN STATEMENTS

The FATAR Block Modification and Scan control statements (PRINT, DROP, KEEP, REPLACE, and SCAN) invoke optional facilities of FATAR to print data from TAPEIN, to scan TAPEIN data blocks for certain data or types of data, and to modify TAPEIN data blocks before they are written to TAPEOUT. The RENAME statement is also a Modification statement, but is used to rename files as they are copied to the output tape.

There may be as many Modification/Scan statements as required. More than one statement may refer to the same block on TAPEIN; if so, the statements are executed in the order in which they appear in the control statement input. An exception to this is the PRINT Statement, since each data block will be printed only once; the **last** PRINT Statement that applies to a given data block is the one that will control its printing.

All operands must be specified on the Modification/Scan control statements in the order shown in the statement formats below. However, it is not necessary to indicate the absence of optional operands with a comma. Because many of the operands are common on these statements, the operands are described separately in Section 23.4, except where special notes apply to a particular statement.

These statements can be confusing to use properly. The FATAR examples in Section 26 contain examples of the use of each kind of Modification/Scan statement. You may wish to review them as you read the following descriptions.

23.3.1 PRINT STATEMENT

PRINT STATEMENT

PRINT **ALL**

,LF=ALL|n**H|**n**|**n**T**

,B=ALL|b_1**|**b_1**-**b_2

,L=l

,CHAR|HEX|DUMP

The PRINT statement requests that data blocks from TAPEIN be printed after other Modification Statements have been applied. By default, the length and format of the printout are controlled by the PRTLEN=, CHAR, HEX, and DUMP operands on the ANALYZE/COPY statement (that themselves default to 80 bytes in character format), but this can be overridden by the L= and CHAR/HEX/DUMP operands on the PRINT statement itself.

Note that the print length refers to the number of characters or bytes to print from each block, even if the blocks are being deblocked into logical records. If the length to print is longer than one logical record, the printout will be formatted into logical records.

The PRINT ALL statement causes all blocks in all files on TAPEIN to be printed, and cannot have any other operands on it. In this case, the length and format are controlled by the above-mentioned ANALYZE/COPY operands.

The ability to use the PRINT statement may be restricted by security rules; see Section 21.6.

Examples:

```
PRINT ALL
PRINT LF=ALL,B=1-5
PRINT LF=3,B=4-7,L=400,DUMP
PRINT LF=1T,B=ALL,HEX
```

23.3.2 DROP STATEMENT

DROP STATEMENT

DROP **DCK**

,LF=ALL|n**H**|n|n**T**

,B=ALL|b_1|b_1-b_2|$b_1.r_1$|$b_1.r_1$-$b_2.r_2$

The DROP statement causes one or more blocks or records to be dropped (not copied from TAPEIN to TAPEOUT). If the DCK operand is specified, the drop will be done only if the affected block had a permanent data check when read from TAPEIN. If B=ALL is given, the entire file referenced by LF= will be dropped from TAPEOUT, including all tapemarks and labels (to drop all data from a file without dropping the file itself, specify "B=1-2147483647").

Dropping individual records (the "B=$b_1.r_1$" or "B=$b_1.r_1$-$b_2.r_2$" formats), is effective only if the blocks on TAPEIN are being deblocked into records. If all of the records in a block are individually dropped, the entire block will be dropped.

***Note**:* Unless the KEEP operand is specified on the ANALYZE/COPY statement, the default operation of FATAR is to drop (not copy to TAPEOUT) any block that has a permanent data check on TAPEIN. It is not necessary to include a DROP DCK,LF=ALL,B=ALL statement to accomplish this. Including such a DROP statement will cause FATAR to unnecessarily generate a message for every block that it **did not drop**.

The DROP statement is ignored for image copies (LABELS=IMAGE).

Examples:

```
DROP  LF=3,B=ALL
DROP  LF=1,B=1-2
DROP  LF=3,B=5.6-6.2
DROP  DCK,LF=7,B=ALL
```

23.3.3 KEEP STATEMENT

KEEP STATEMENT

KEEP **DCK**
,LF=ALL|n**H|**n**|**n**T**
,B=ALL|b_1**|**b_1**-**b_2
,L=l
,P=p
,J=R

The KEEP statement causes blocks from TAPEIN to be copied to TAPEOUT, even if they would be dropped because of a DROP statement, or because of a permanent data check (the DCK operand is required to keep a data check block). Note that if you want to keep **all** such data check blocks, use the KEEP operand on the ANALYZE/COPY statement instead of a KEEP statement.

If the optional L= operand is given, the KEEP statement may be used to change the length of a block as it is being copied from TAPEIN to TAPEOUT. L= specifies the new block length and P=, if given, specifies location in the new block (relative to 1) where the old data will be placed. J=R requests that the old data be right-justified in the new block, with the rightmost byte of the old data block positioned at the location specified by P=; if J=R is omitted, the data is left justified, with the leftmost byte placed at the P= location. These combinations allow full control over repositioning of the data. Any extra bytes added at the front or end of the new block will be set to spaces (blanks). Whenever the length of a block is changed, deblocking of that block into logical records will be terminated.

Examples:

```
KEEP  LF=1,B=ALL
KEEP  LF=ALL,B=1-5
KEEP  DCK,LF=1,B=ALL,L=5000
```

23.3.4 REPLACE STATEMENT

REPLACE STATEMENT

REPLACE DCK

,LF=ALL|n**H**|n|n**T**

,B=ALL|b_1|b_1-b_2|$b_1.r_1$|$b_1.r_1$-$b_2.r_2$

,P=p

,V=string

,S=string

,D=string

,ALL

The REPLACE statement is used to modify data within blocks being copied from TAPEIN to TAPEOUT, and also to scan for blocks or records containing a specified data string (even if TAPEOUT is not present). If TAPEIN data blocks are being deblocked, the REPLACE Statement operates on a logical record basis.

If D= is specified **without V= or S=**, the data string specified by D= will be placed in every affected data record or block at the location given by P=.

If D= is specified **with V=**, the data in every affected data record or block will be verified for the current contents specified by V= before the data from D= is placed there (again, the location within the block or record is given by P=). This allows you to be sure the proper data field is being replaced, or to search a set of records for the proper one to modify.

If D= is specified **with S=**, every affected record or block will be scanned (starting at the location specified by P=, default 1) for the data string specified by S=. If found, the replacement data given by D= will be stored at the location where the S= data was found. Normally, only one such replacement will be done in each record or block, but if the parameter ALL is also given, the scan will continue until the end of the record/block.

If V= or S= is given **without D=**, verification or scanning will take place as described above, but no data modification will be done. This allows the REPLACE statement to be used to scan data for specific data strings, since any block on which REPLACE successfully operates will be eligible for printing.

Examples:

```
REPLACE LF=ALL,B=ALL,S=C'ORIGINAL',D=C'NEW DATA',ALL
REPLACE LF=3,B=ALL,P=10,V=X'1302',D=X'1503'
REPLACE LF=ALL,B=5.2-20.6,S=C'FIND THIS STRING'
```

23.3.5 SCAN STATEMENT

SCAN STATEMENT

SCAN DCK
,LF=ALL|n**H|**n**|**n**T**
,B=ALL|b_1**|**b_1**-**b_2**|**$b_1.r_1$**|**$b_1.r_1$**-**$b_2.r_2$
,L=l
,P=p
,T=t

The SCAN statement causes specified data fields to be validated for specific type of content. Scans are performed on a logical record basis if records are being deblocked. The field to be scanned is at the location specified by P= (relative to 1) for the length specified by L=. If T=Z is specified, the field is checked for valid zoned decimal characters including decimal sign characters. If T=P is specified, the field is checked for valid packed decimal characters. For any other value of T=, there must be a corresponding TABLE Statement with the same T= value, and the SCAN statement will validate the presence or absence of the characters specified there. A message will be issued for every field that fails the SCAN test and the failing block will be printed.

Examples:

```
SCAN  LF=1,B=ALL,L=5,P=22,T=P
SCAN  LF=ALL,B=ALL,L=7,P=9,T=Z
SCAN  LF=3,B=1-20,L=5,P=20,T=A    (see TABLE example)
```

23.3.6 TABLE STATEMENT

TABLE STATEMENT

TABLE **T=**t

,R

,D=string

The TABLE statement is used to define a set of characters for SCAN statement operations. T= must specify a unique letter (except for P and Z, which are reserved) for naming the table, so up to 24 tables can be defined. Each table can contain up to 256 characters, specified by the D= operand. The TABLE statement must precede any SCAN statement that references it. If a field being scanned contains any character not in the table, it will be flagged and its block printed. If the optional operand R is included, this operation is reversed, so that if the field contains any character that is in the table, it is flagged.

Examples:

```
TABLE T=A,D=C'0123456789ABCDEF'  printable hex
TABLE T=A,D=C' ADFQR',
         X'FF7F',C'269'
```

23.3.7 RENAME STATEMENT

RENAME STATEMENT

RENAME UNCOND
,LF=ALL|n
,NEWN=newname**|NEWI=**newindexmask

When an output tape is being created, RENAME can be used to change the name of the file(s) created on the output tape. The LF= operand specifies to which files on the input tape this RENAME statement applies; it can be for a single file, or for all files on the tape (in the latter case, the NEWI= operand is usually used**). No comments are allowed on the RENAME statement.**

NEWN= specifies a full replacement data set name, up to 44 characters. If the name ends in a GDG relative generation number, e.g., NEWN=PROD.MAST(+1), a LOCATE is done to get the proper absolute generation number. Note that if NEWN= is used with LF=ALL, all copied files on the output tape will have that name.

NEWI= specifies that the output data set name is to be constructed by adding or replacing one or more index levels in the original name (from the input tape); replacement index levels do not have to be the same length as the original indexes they replace. In the simplest case, FATAR will use each index level specified in NEWI in place of the corresponding index in the original name. Any remaining index levels at the end of the name are copied unchanged. This can easily be used to change the first indexes of the name.

For example, if copying data set A.B.C.D,

`NEWI=D`	results in	`D.B.C.D`	(first index replaced)
`NEWI=DD.E`	results in	`DD.E.C.D`	(first 2 indexes replaced)

If a period is specified without any preceding characters, FATAR will copy one original index level from the input data set name to the output. This allows you to easily modify indexes in the middle of the name.

For example, if copying data set A.B.C.D,

`NEWI=..E`	results in	`A.B.E.D`	(third index replaced)
`NEWI=FF..G`	results in	`FF.B.C.G`	(first and fourth indexes replaced)

If + is specified before a new index level, FATAR will insert that new index into the output data set name at that point. If ++ precedes the new index, it will be added to the end of the name. If − is specified, the next input index level will be dropped from (not copied to) the output name.

For example, if copying data set A.B.C.D,

`NEWI=+F`	results in	`F.A.B.C.D`	(new first index added)
`NEWI=++F`	results in	`A.B.C.D.F`	(new last index added)
`NEWI=..+F`	results in	`A.B.F.C.D`	(new third index added)
`NEWI=..-`	results in	`A.B.D`	(third index dropped)
`NEWI=Q.-.+E`	results in	`Q.C.E.D`	(combination)

Note that, except for the ++ option, every period in the NEWI= mask corresponds to a period (one index level) in the original (input) data set name. FATAR will check the resulting new name to insure it meets IBM standards.

If the NEWI= value ends in a GDG relative generation number, e.g., NEWI=..NEWMAST(--2), that relative number will be added to the end of the newname, and a LOCATE done to get the proper absolute generation number.

NEWI= is a convenient way to rename every file on a multi-file input tape as it is copied to the output tape, while using some index levels from the original name and replacing other indexes or adding new indexes.

Normally, the NEWI= operand is honored only if FATAR knows the full, accurate data set name of the input file being copied; this is usually true only if the FATS/FATAR tape management interface is enabled (see Section 21.5). To use NEWI= when the only data set name is the truncated 17-character name from the tape labels, specify the UNCOND operand; NEWI= will be applied to that truncated name except that if the truncated name starts with a period, it is discarded. If used, UNCOND must be the first operand on the RENAME statement.

RENAME is ignored when LABELS=IMAGE has been specified.

The ability to use the RENAME statement may be restricted by security rules; see Section 21.6.

23.4 MODIFICATION/SCAN OPERANDS

The following are details of the keyword parameters that may appear on the Modification/Scan Statements in the preceding sections. Consult the definitions of each statement to see which parameters are valid on each statement, and in what form.

ALL On a REPLACE statement containing the S= operand, specifies that all occurrences of the scan string in the current block or record are to be replaced. If omitted, only the first occurrence of the scan string is replaced.

On a PRINT statement, specifies that all data blocks are to be printed, using the formatting requested by the PRTLEN= and CHAR/HEX/DUMP operands on the ANALYZE/COPY statement.

B=ALL
B=b_1
B=b_1-b_2
B=$b_1.r_1$
B=$b_1.r_1$-$b_2.r_2$

Specifies to which blocks within the file specified by LF= the statement applies.

B=ALL requests all blocks in the file.

B=b_1 causes it to apply only to one specified block.

B=b_1-b_2 specifies a range of blocks (b_1 less than b_2).

B=$b_1.r_1$ indicates a specific record within a specific block (record deblocking must be in effect).

B=$b_1.r_1$-$b_2.r_2$ causes the statement to apply to a range of records from record r1 in block b1 to record r2 in block b2 (b_1 and b_2 may be the same).

b_1, b_2, r_1, and r_2 are all unsigned decimal numbers from 1 to 2,147,483,647 (without commas). Blocks within a file and records within a block are numbered starting with 1.

CHAR
HEX
DUMP

On a PRINT statement, specifies an overriding print format (character, hexadecimal, or dump) for this PRINT statement only. If omitted, the print format specified on the ANALYZE/COPY statement will control the format for this PRINT statement.

DCK Specifies that FATAR is to perform the indicated operation only if the block affected had a permanent data check when read from TAPEIN.

J=R On KEEP statements only, specifies that the old data block is to be right-justified in the new data block (if not specified, the old data block is left-justified).

L=l For PRINT, specifies an overriding print length.

For KEEP, specifies the new length of the block.

For SCAN, specifies the length of the field to verify.

"l" must be an unsigned decimal number.

LF=ALL
LF=n
LF=n**H**
LF=n**T**

Specifies the logical file on the input tape to which this statement applies. "n" is a decimal number giving the logical file number. If the input tape is labeled, this is the file sequence number (1-65535), the same value you would specify in the LABEL=n parameter in JCL (except that LABEL= only accepts values up to 9999). Specifying "n" alone will cause the statement to apply to the equivalent data file. Appending an "H" or "T" after the file number causes it to apply to the header or trailer label file, respectively, preceding or following that data file. If the input tape is not labeled, or if labels are not being processed (LABELS=NO), then "n" is the physical file number (1-196605) relative to the beginning of the tape (file 1).

LF=ALL specifies that this statement applies to all files of the input tape.

***NOTE**:* For compatibility with Version 3.0 of FATAR, the parameter F=n will be accepted in place of LF=. However, in this case, "n" specifies the physical file number only (1-196605) regardless of the label status of the tape, and header and trailer label files must be counted. Use of F= is not recommended for new applications.

P=p

For SCAN and REPLACE, specifies the starting location for the operation; if FATAR is deblocking logical records, it specifies a location within a record. For KEEP, specifies the starting location (or ending location if J=R) of the old data block within the new block.

"p" must be an unsigned decimal number. P=1 is the default and specifies the first byte of the record or block. For deblocked variable format records, the first data byte is at location 5, past the 4 byte RDW (Record Descriptor Word).

T=t

On a TABLE statement, defines the name of a table for reference by a SCAN statement. "t" must be a unique letter other than P and Z, which are reserved.

On a SCAN statement, specifies the type of scan to be performed:

T=Z -- scan for zoned decimal numbers

T=P -- scan for packed decimal numbers

T=t -- scan according to user table "t"

V=string
S=string
D=string

Defines a data string on a REPLACE or TABLE statement. The data may be in EBCDIC characters or hexadecimal bytes, or a combination of both, using this format:

C'chardata**'** - defines an EBCDIC string up to 255 bytes in length.

X'hexdata**'** - defines a hexadecimal string up to 255 bytes in length.

A string consisting of both character and hexadecimal data may be defined by simply placing the C and X strings after one another, separated by commas. For example,

```
D=C'ABC',X'43C721',C'XYZ'
```

defines a 9-byte string consisting of the 3 separate strings concatenated together.

Longer strings obviously cannot fit on a single control statement record, so a data string is continued onto another record by closing the string with a quote and comma followed by a blank, and starting on the next statement (in any column) with X' or C'. For example,

```
V=C'THIS IS A VERY LONG STRING THAT IS GOING TO BE ',
 C'CONTINUED ON THE NEXT RECORD'
```

This page intentionally left blank.

24.0 FATAR SPECIAL CONSIDERATIONS

This section will examine special considerations for the use of FATAR, primarily recovering from special sorts of errors.

24.1 VARIABLE SPANNED RECORDS

S002 ABENDS

S002 abends on variable spanned format tapes (RECFM=VBS or VS) can be very frustrating because of the difficulty in diagnosing and correcting the errors. By far the most common use of VBS format is with SMF/RMF data, since their format is variable with few limitations on the maximum logical record size, so a S002 abend on those tapes may threaten an installation's accounting and performance data.

S002 abends usually indicate an improperly spanned logical record, due to some logical or physical error when creating the tape, or possibly due to blocks dropped when copying with FATAR because of permanent data checks. FATAR is one of the few programs that can detect and correct these spanning errors, but it is a process requiring several passes of the tape and some human analysis.

VARIABLE SPANNED FORMAT

Variable spanned tapes have the characteristic that logical records may "span" across two or more physical blocks. Because of this, a logical record may be larger than the blocksize of the file. Because VBS format attempts to fill each physical block to capacity, the last logical record in each block, regardless of size, will almost always span to the next block, so almost every block begins and ends with a spanned record.

When a record is broken into pieces for spanning, each piece is referred to as a segment. There are flags in the RDW (record descriptor word, preceding each variable length record) that indicate "first segment", "middle segment", or "last segment" of the total logical record. A "middle segment" exists only when a logical record spans three or more blocks.

***Note**:* IBM uses VBS format for SMF data on disk because the maximum SMF record size (32760) exceeds the blocksize typically used for the SMF data sets. However, when SMF data is copied to tape, there is no reason that it must remain in VBS format. Since tapes can easily support blocks up to 32767 bytes in length, Innovation recommends that SMF data be converted to RECFM=VB (not spanned) during the copy to tape so that spanning errors on SMF data can be avoided altogether. Many utilities that are used for the SMF data copy can do this conversion. Unfortunately, even on a tape-to-tape copy, FATAR cannot do this type of conversion.

FATAR SPANNED FORMAT CHECKING

FATAR automatically performs spanning checks on VS or VBS tapes, detecting missing first or last segments. When FATAR detects a missing segment, it prints the message FATS069 START (or END) OF SEGMENT MISSING. The block and record numbers printed to indicate where the error was detected, but the actual problem is usually in the preceding block. End of segment missing indicates that FATAR found the start of a new logical record before finding the end of the preceding logical record, that started in one of the immediately preceding blocks.

Start of segment missing indicates that FATAR found a middle or end segment not preceded by a start segment, which again should have been in the preceding block. One situation FATAR cannot detect is a missing middle segment where the start and end segments exist; it might even be that those start and end segments do not even belong together because of lost intermediate blocks, but there is no way to detect that. These errors will not usually cause S002 abends, but may cause data errors in the application programs reading them. If you can identify the records causing the error, FATAR may still be used to correct this problem.

CORRECTING S002 ABENDS

To correct these problems with FATAR, follow this procedure:

1) Analyze the tape once with FATAR (no TAPEOUT) and note which blocks and records FATAR identifies as having spanning errors (FATS069 messages).

2) Analyze the tape again, printing the complete blocks for at least three blocks before and after each block identified by FATS069, using this input:

```
ANALYZE  PRTLEN=32760,DUMP
PRINT   LF=f,B=b1-b2
```

substituting the proper file number (f) and a range of block numbers (b1-b2). Include multiple PRINT statements if multiple FATS069 messages occurred.

3) Examine the printout to see where various logical records start and end. FATAR will print each physical block broken down into logical records; it prints the block number and then numbers the logical records (or segments) in the block. You can see the spanning flags in the printout in byte 3 of each segment:

   00 -- complete logical record (not spanned)

   01 -- start segment

   02 -- end segment

   03 -- middle segment

4) If you received START OF SEGMENT MISSING, then the middle and end segments remaining must be deleted. Starting at (and including) the block and record number indicated in the FATS069 message, look at each segment until you find the end segment (byte 3 = x '02'). Write down the block number and record number of each segment you examine (including the end segment) since each must be deleted.

5) If you received END OF SEGMENT MISSING, then start at the record preceding the one identified in the FATS069 message and look at each record going backwards until you find the start segment (byte 3 = x '01'). Write down the block and record number of each segment examined in order to delete them. Do not include the record from the FATS069 message.

6) Run FATAR again with a TAPEOUT DD statement in order to copy and correct the spanned tape, using this input:

```
COPY VERIFY
DROP LF=f,B=b.r
```

substituting the proper file number (f), block number (b) and record number (r) to be deleted. Include a DROP statement for each segment you identified above. The dropped segments should eliminate the spanning errors, and the verify option will cause the output tape to be analyzed to be sure. No FATS069 messages should occur during the verification phase.

24.2 RECOVERING DATA FROM OVERWRITTEN TAPES

OVERWRITTEN TAPES

A special recovery problem exists when a tape that still contains important data is accidentally used for output, resulting in an overwritten tape. Although tape management systems usually prevent the accidental overwriting of a tape that is not in scratch status, sometimes a tape will expire and be used for output while the original data is still needed, probably because of incorrect assignment of tape expiration dates.

Any data that is physically overwritten is gone, unfortunately; but if the new data file is shorter than the original data file, it is possible to recover the remains of the original data and copy it to another tape. This recovered data may or may not be useful, depending on the format of the data and its intended use.

To recover the remains of the partially overwritten data file, you must position FATAR to the beginning of that data, allowing FATAR to read and copy the remaining data. Since the header labels of the file are also overlaid, you will need some knowledge of the original data to be able to specify proper DCB information. Examples in Section 26.2 shows how you might do this.

Because the first block of the remaining original data is probably partially overwritten, you will almost always get a data check on that block. Since FATAR will automatically discard that block, this can be ignored; the remaining blocks should be readable. Occasionally, the error on that first block will make the tape drive believe that a more serious error exists and refuse to read further. If this recurs on several different tape drives, the tape is probably not recoverable.

If the tape originally contained multiple data files, it may be that some of the original files are intact with complete header and trailer labels, yet they cannot be opened normally because of the overlaying data earlier on the tape. You can use FATAR to completely and automatically copy those files. Examples in Section 26.2 shows an example of this function. The tape layouts below will help you understand the value of "n" to be used, but it may require some experimentation to get the right value.

EOD MARKS

On tapes written on cartridge drives (except 3480), the drive writes an EOD (End-of-Data) mark after the last written data on each tape volume. The EOD is designed to make it more difficult for an end-user to read past the end of a data file to which he is authorized and read residual unauthorized data from the tape. But it also makes the job of a legitimate recovery more difficult. When the tape is overwritten, an EOD is written immediately after the last overwriting file. Normal access methods cannot get past the EOD.

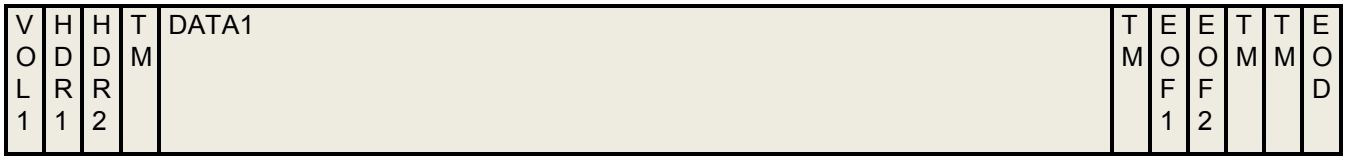
Using techniques that Innovation has obtained from IBM and other tape drive vendors, FATAR will attempt to bypass the EOD mark and recover the data beyond it. The technique used may depend on the manufacturer of the drive. In some cases the recovery may not be successful. In others, it may work but is forced to skip over some of the original data before beginning the recovery.

The ability of FATAR to read data past an EOD mark may be restricted by security rules; see Section 21.6.

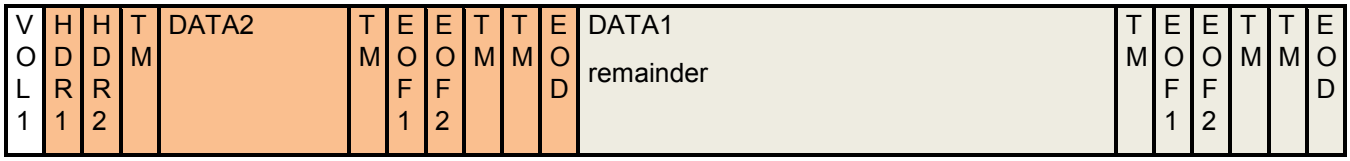
EXAMPLES OF OVERLAYING DATA

In these examples, **TM** represents a tape mark, a hardware end-of-file. **EOD** represents a hardware End-of-Data mark, that will exist only on tapes created on IBM 3490E and 3590 (Magstar) drives and compatible drives.

If the tape originally contained a single data file, it looked like this:

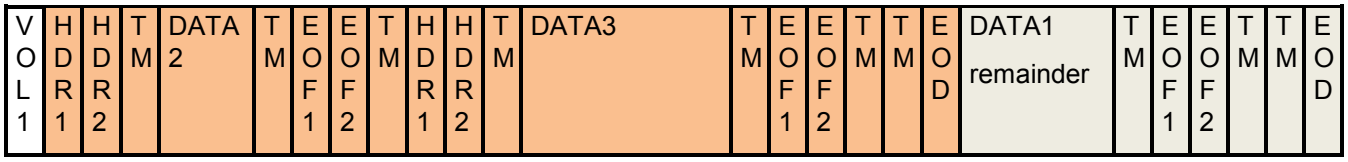


If it is overlaid with a single data file shorter than the original it will look like:



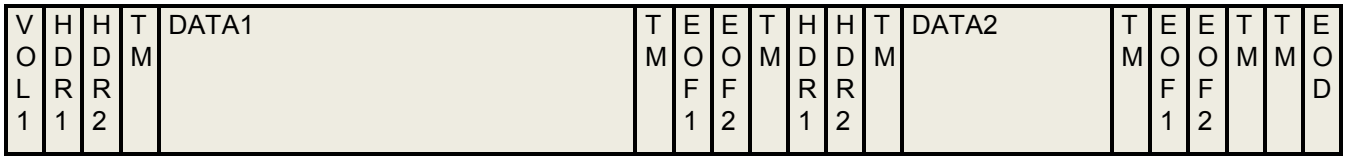
There are 4 tape marks before the remainder of the original data file, so use LABEL=(5,BLP).

If the same tape is overlaid with a two short data files it will look like:

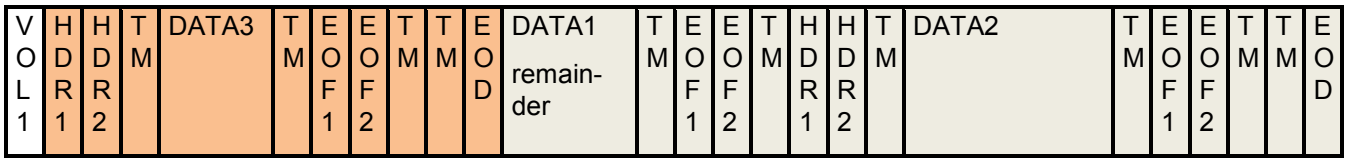


There are 7 tape marks before the remainder of the original data file, so use LABEL=(8,BLP) to recover the remainder of the original file 1.

If the tape originally contained two data files, it looked like this:



If it is overlaid with a single data file shorter than the first original file it will look like:



There are 4 tape marks before the remainder of the original first data file, so use LABEL=(5,BLP) to recover it. There are 6 tape marks before the header labels of the second data file, so LABEL=(7,BLP) is used to recover it.

RECOVERY TECHNIQUES

If you are recovering from an overwritten tape that supports EOD marks (all cartridge drives except 3480s and compatibles), recovery is fairly easy. Specify LABELS=EOD on the ANALYZE/COPY statement and FATAR will do most of the work of recovering all the residual data on the tape.

LABELS=EOD instructs FATAR to read up the tape until an EOD mark is encountered. Then it reads the data following that EOD mark. The first file after the EOD mark is (usually) partially overwritten, so you have to specify a data set name and DCB information on the TAPEOUT DD statement so that copied data can be used. If there are additional files on the tape beyond that first overwritten file, they will also be copied, using the data set names and DCB information in the header labels of those files. No special LABEL= parameters are required on TAPEIN but TAPEOUT should specify or default to LABEL=(,SL). See example 22 in Section 26.2

For round tapes and 3480 cartridges, recovery is a 2-step process:

a) First, recover the partially overwritten data file. Specify LABELS=NO on the ANALYZE/COPY statement and LABEL=(n,BLP) on the TAPEIN DD to position FATAR to the start of that file, where "n" is the physical file number of that data. You can calculate "n" by counting the number of tape marks preceding that data and adding 1. You can see in the example tape layouts on the previous page that ”n" is usually 5 if there is only one overwriting file. See example 23 in Section 26.2.

b) If there were additional files on the tape beyond the partially overwritten file, you recover them by specifying LABELS=YES on the ANALYZE/COPY statement and LABEL=(n,BLP) on the TAPEIN DD to position FATAR to the header labels of the first additional file, where "n" is the physical file number of those header files. "n" is usually 7 if there is only one overwriting file. See example 24 in Section 26.2.

c) If you are unsure of the format or size of the original or overlaying files, you can run a FATAR ANALYZE/COPY on the tape with NUMFILES=999 and LABELS=NO; this will continue reading past the 2 consecutive tape marks at the end of the overlaying data file and show you what remains on the tape.

24.3 BENEFITS OF USING FATS/FATAR WITH A VTS

USING FATS/FATAR WITH A VIRTUAL TAPE SYSTEM

Many customers are migrating to virtual tape devices. FATS and FATAR are useful on a virtual tape system for performing many tasks:

1. Investigating a volume, to discover its label type, file count, DCB characteristics, etc., and providing a compact summary of the characteristics of all files on the volume.
2. Examining the data on a volume.
3. Verifying that certain data fields contain valid data.
4. Creating a backup copy of any volume (or multi-volume set of tapes), even if multiple files exist on the volume. FATAR can be used to make a backup physical copy of a virtual tape volume.
5. Verifying that a data set is properly formatted (every block is checked against its DCB information).
6. Correcting invalid data by creating a copy of the input volume(s) with the bad data corrected or dropped.
7. Creating a good copy of a volume that was not properly closed (such as during a system failure).
8. Replacing volumes with small blocksizes with copies using a larger blocksize (reducing elapsed and CPU times of applications that read that data).
9. Making an "image copy" (an exact bit-for-bit copy) of a tape volume.
10. Detecting and correcting invalid variable spanned records. An example of this procedure can be found on the Innovation FTP site, http://www.fdr.com/ftp/ftp.cfm, in the file

/Public/Download/FATS_FATAR_V49/ FATAR example - correct invalid variable spanned records.pdf.

25.0 FATAR REPORT EXAMPLES

In this example, FATAR was executed with no control statements (using all default values). The input was a standard label 3480 cartridge containing 2 files; device type 3480XF is a drive capable of IDRC compaction, but this tape is not compacted. There is a data check on a block in the middle of the second file. 80 bytes of the block with the data check are displayed, as well as for the block before and after the data check block.

```
FATS014 NO CONTROL CARD DATASET. DEFAULTS ASSUMED
FATS071 TAPE BUFFER SIZE IS 65535 BYTES
                                    CHARACTERISTICS OF THE TAPE TO BE ANALYZED
                                             UNIT  DEVICE   SERIAL
                                             0391  3480XF   800432
                                              FATAR DETAIL REPORT
 BLOCK  LNGTH/  MESSAGE/               1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
 NUMBER DISPL  BLOCK TYPE                 (COLUMN GRID IS VALID ONLY FOR CHARACTER FORMATTED DATA)
* * * * * * *  START FILE       1H
      1    80  INPUT LABEL             VOL1800432
               ***VOLUME LABEL***           VOLSER=800432 OWNER=
      2    80  INPUT LABEL             HDR1RAY.SCR010       80043200010001       86192 000000000000IBM OS/VS 370
               ***HEADER LABEL***           DSNAME=RAY.SCR010        SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0001
                                            CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT=    000000
      3    80  INPUT LABEL             HDR2U3276000000000SPSN    /COPY1          14850
               ***HEADER LABEL***           RECFM=U    LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
                                            DENSITY=0 TRTCH=   CREATING JOB/STEP=SPSN    /COPY1
* * * * * * *  END OF FILE      1H -- FILE CONTAINED       3 BLOCKS
* * * * * * *  START FILE       1
* * * * * * *  END OF FILE      1  -- FILE CONTAINED     486 BLOCKS
* * * * * * *  START FILE       1T
      1    80  INPUT LABEL             EOF1RAY.SCR010       80043200010001       86192 000000000486IBM OS/VS 370
               ***END-OF-FILE***            DSNAME=RAY.SCR010        SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0001
                                            CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT=    000486
      2    80  INPUT LABEL             EOF2U3276000000000SPSN    /COPY1          14850
               ***END-OF-FILE***            RECFM=U    LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
                                            DENSITY=0 TRTCH=   CREATING JOB/STEP=SPSN    /COPY1
* * * * * * *  END OF FILE      1T -- FILE CONTAINED       2 BLOCKS
* * * * * * *  START FILE       2H
      1    80  INPUT LABEL             HDR1RAY.SCR501       80043200010002       86192 000000000000IBM OS/VS 370
               ***HEADER LABEL***           DSNAME=RAY.SCR501        SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0002
                                            CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT=    000000
      2    80  INPUT LABEL             HDR2U3276000000000SPSM    /COPY1          14850
               ***HEADER LABEL***           RECFM=U    LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
                                            DENSITY=0 TRTCH=   CREATING JOB/STEP=SPSM    /COPY1
* * * * * * *  END OF FILE      2H -- FILE CONTAINED       2 BLOCKS
* * * * * * *  START FILE       2
      8 17718  ***PERMANENT DATA CHECK -- RETRIES= 0, CLEANING ACTIONS= 0, ROR RETRIES= 0
               DATA TRANSFERRED                 ::      :      :      :      :      :      :      :      :   ::
               BLOCK DROPPED
      7 47626  BLOCK PRECEEDING D/C     :::: :     : :  :    :  :: :00000050:   :  ::: :  :: :00000050:   :  ::: :  :: :
      9 44970  BLOCK FOLLOWING D/C      :::d : :   : : ::    : :::::000016200000161ICCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
* * * * * * *  END OF FILE      2  -- FILE CONTAINED     580 BLOCKS
* * * * * * *  START FILE       2T
      1    80  INPUT LABEL             EOF1RAY.SCR501       80043200010002       86192 000000000580IBM OS/VS 370
               ***END-OF-FILE***            DSNAME=RAY.SCR501        SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0002
                                            CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT=    000580
      2    80  INPUT LABEL             EOF2U3276000000000SPSM    /COPY1          14850
               ***END-OF-FILE***            RECFM=U    LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
                                            DENSITY=0 TRTCH=   CREATING JOB/STEP=SPSM    /COPY1
* * * * * * *  END OF FILE      2T -- FILE CONTAINED       2 BLOCKS
* * * * * * *  START FILE       3H
* * * * * * *  END OF FILE      3H -- FILE CONTAINED       0 BLOCKS
FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE
                                                  FINAL TOTALS
         FILES READ  BLOCKS READ  BYTES READ   FEET READ  TEMP ERRS  PERM ERRS  FILES WRITTEN  BLOCKS WRITTEN
               2         1066      40367286        96         0          0           0              0
```

```
FATAR                          TAPE SUMMARY FOR TAPE VOLUME -800432- AT DENSITY   38000 BPI                  06/01/2001
PHYS DATASET NAME       FILE  FIL#                    REC- LRECL CREATING     BLOCKS   BYTES  PERM ---BLOCKSIZES---- EST.
FILE  (LAST 17 CHARS)  SERIAL VOL#  CRDATE   EXPDATE   FM  BLKSZ JOB&STEP SEC  READ     READ  TEMP  MIN   AVG   MAX  FEET
---- ----------------- ------ ---- -------- -------- ---- ----- -------- --- -------- ------- ---- ----- ----- ----- ----
   2 RAY.SCR010        800432 0001 1986/192   00/000 U    00000 SPSN      0      486     17M    0    24 34213 48092   39
                              0001                        32760 COPY1                          0
   5 RAY.SCR501        800432 0002 1986/192   00/000 U    00000 SPSM      0      580     24M    0    24 40931 49060   56
                              0001                        32760 COPY1                          0
                                                                                                                    ____
                  HIGHEST EXPIRATION ====>   00/000           TOTALS ====>     1066     40M    0                    95
```

In the next example, a 3590 (Magstar) tape volume is copied to another Magstar. The input tape is hardware compacted, so compaction (IDRC) statistics are displayed. Lengths are to be displayed in metric, and 64 bytes from the first 3 blocks of every file are to be displayed in dump format (hex and EBCDIC).

```
FATAR CONTROL CARDS

   1--  COPY METRIC,BLP,PRTLEN=64,DUMP                                    00200026
   2--  PRINT LF=ALL,B=1-3                                                 00210026
FATS071 TAPE BUFFER SIZE IS 65535 BYTES
FATS072 TAPEIN DATA WILL BE COPIED TO TAPEOUT

                                  CHARACTERISTICS OF THE TAPE TO BE ANALYZED
                                     UNIT  DEVICE   SERIAL  MFR CARTRIDGE
                                     03A0  3590     900021  IBM  10 MB

                                            FATAR DETAIL REPORT
 BLOCK  LNGTH/  MESSAGE/
 NUMBER DISPL  BLOCK TYPE
* * * * * * *  START FILE       1H
      1    80  INPUT LABEL              VOL1900021
               ***VOLUME LABEL***            VOLSER=900021 OWNER=
      2    80  INPUT LABEL              HDR1BAB.LINK         90002100010001       9700200000000000000IBM
               ***HEADER LABEL***            DSNAME=BAB.LINK          SERIAL=900021 VOLSEQ#=0001 FILESEQ#=0001
                                             CRTDT= 99002 EXPDT=000000 SECURITY=0 BLOCKCOUNT=    000000
      3    80  INPUT LABEL              HDR2V061640616000BABFATTS/IEBCOPY P    S  9C8F10
               ***HEADER LABEL***            RECFM=VS   LRECL=06160 BLKSIZE=06164 BUFOFF(ANSI)=
                                             DENSITY=0 TRTCH=P  CREATING JOB/STEP=BABFATTS/IEBCOPY
* * * * * * *  END OF FILE      1H -- FILE CONTAINED       3 BLOCKS
* * * * * * *  START FILE       1
      1    64  PRINT REQUESTED 00000006
               003C0000 00CA6D0F 02001800 0000C000  00001814 3030200F 00007FF8 08B3000F
......_.....................8....
       +00033 E5A20000 22500000 00020061 00020000  00800002 3F09A71A 947C0000           V....&...../.............@..
      2   284  PRINT REQUESTED 00000006
               01180000 02000000 FF000000 8F009000  047BDD88 58F47A40 000000AF 000C0148  .................#...4:
........
       +00033 000E08FA 58F47A40 00000601 00000627  0004023F 00000000 00000000 00000000  .....4:
........................
      3  6080  PRINT REQUESTED 00000006
               17BC0000 00000000 00000000 00080100  C1C4C4E2 C4404040 00EAC1C2 C1404040  ................ADDSD   ..ABA
       +00033 404000B2 412C00B3 02000000 000002C2  00408002 A00002A0 88020101 0000C1C3    .............B.
............AC
* * * * * * *  END OF FILE      1  -- FILE CONTAINED   21127 BLOCKS -     136 IDRC (COMPACTED) BLOCKS
* * * * * * *  START FILE       1T
      1    80  INPUT LABEL              EOF1BAB.LINK         90002100010001       9700200000000021127IBM
               ***END-OF-FILE***             DSNAME=BAB.LINK          SERIAL=900021 VOLSEQ#=0001 FILESEQ#=0001
                                             CRTDT= 99002 EXPDT=000000 SECURITY=0 BLOCKCOUNT=    021127
      2    80  INPUT LABEL              EOF2V061640616000BABFATTS/IEBCOPY P    S  9C8F10
               ***END-OF-FILE***             RECFM=VS   LRECL=06160 BLKSIZE=06164 BUFOFF(ANSI)=
                                             DENSITY=0 TRTCH=P  CREATING JOB/STEP=BABFATTS/IEBCOPY
* * * * * * *  END OF FILE      1T -- FILE CONTAINED       2 BLOCKS
* * * * * * *  START FILE       2H
* * * * * * *  END OF FILE      2H -- FILE CONTAINED       0 BLOCKS
FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE
                                                 FINAL TOTALS
          FILES READ  BLOCKS READ  BYTES READ METERS READ  TEMP ERRS  PERM ERRS  FILES WRITTEN  BLOCKS WRITTEN
COMPACTED %USED
                   1         21127       83081159     ****         0          0           1           21127
IDRC (COMPACTED)             136       43470848       11                                                      48%
5%
                        ****    ****************************************************************
                                ***   3590  AND COMPACTION REDUCED TAPE  REQUIREMENTS BY  48%   ***
                                ****************************************************************
FATS026 * * * *  TAPEIN DATA SUCCESSFULLY COPIED TO TAPEOUT  VOLUME 900023
```

```
FATAR                          TAPE SUMMARY FOR TAPE VOLUME -900021- AT DENSITY  105219 BPI
06/01/2001
PHYS DATASET NAME       FILE  FIL#                   REC- LRECL CREATING     BLOCKS   BYTES  PERM ---BLOCKSIZES----
EST.
FILE  (LAST 17 CHARS)  SERIAL VOL#  CRDATE   EXPDATE  FM  BLKSZ JOB&STEP SEC  READ     READ  TEMP  MIN   AVG   MAX   M.
---- ----------------- ------ ---- -------- -------- ---- ----- -------- --- -------- -------- ---- ----- ----- ----- ---
-
   2 BAB.LINK          900021 0001 1997/002   00/000 VS   06160 BABFATTS  0    21127     83M    0    20  3932  6164
****
                              0001                        06164 IEBCOPY IDRC>    136     43M    0      319639
11
                                                    ________                    ________ ________ ____
____
                HIGHEST EXPIRATION ====>   00/000           TOTALS ====>    21127     83M    0
****
                              IDRC COMPACTION ====>  48%  IDRC TOTALS ====>      136     43M
11
```

Because the data on this input tape was compacted (compressed) by the tape hardware, FATAR gives you several statistics about the compaction. Compaction (called IDRC on some types of cartridge) not only compressed the data, it combines user blocks into larger "superblocks" on the tape, all transparent to the user. In the detail report, the final totals show the number of blocks and bytes written by the user, and also shows the physical blocks and bytes written after compaction. The percentage of tape saved (length of tape required for the uncompacted data vs. the length actually used) is displayed. Since the uncompacted data is not written to tape, the actual length (in meters in this example) is displayed only for the compacted data.

In the summary report, the same information is displayed, plus compaction statistics for each input file.

This page intentionally left blank.

26.0 FATAR EXAMPLES OF USAGE

INTRODUCTION

Examples are provided here to guide you in the use of FATAR. Many of the common uses of FATAR have been included, and most of the control statements and keywords are illustrated here. However, be aware that they are just examples and must be customized to your installation and intended use. Areas of customization will probably include: data set names, unit names for tape, special parameters for tape management systems. The examples also assume that no special STEPLIB DD Statement is required to execute FATAR; this may not be true in your installation.

Because of the many possible combinations of options, control statements, tape label types, etc., it is not practical to create examples that cover all possible FATAR uses. Many examples illustrate more than one aspect of FATAR use, e.g., a particular tape label type and use of some FATAR keyword. This does not mean that the two must always go together unless the accompanying text says so.

Some examples use LABEL=(,BLP) on the input tape in order to be able to mount a labeled tape without knowing its volume serial or data set names. Sometimes this is simply a convenience, but for some operations, such as printing the labels from an unknown tape, it is essential. In some installations, system parameters prevent the usage of BLP by most users; in this case, the FATAR option BLP on the ANALYZE/COPY Statement can be substituted. See Section 21.6 for security considerations for the use of BLP.

Some examples illustrate the use of the CAT= operand to catalog output data sets and the RENAME statement with NEWI= to rename copied data sets. These may not work as shown unless you have the FATAR tape management interface (Section 21.5) enabled, to allow FATAR to get the full 44-character data set name of the input data set.

ISPF DIALOG SUPPORT

Most of the FATAR functions can be performed using the FATAR ISPF Interface. See Section 93.2 for further details.

26.1 INDEX TO EXAMPLES

INDEX TO EXAMPLES

The first-time or infrequent user of FATAR should review the examples below in order since they frequently build upon one another in illustrating the usage and effects of various parameters. For your convenience, however, here is a list of the examples provided to aid in quickly finding the one you need. It shows the title of the example, the ANALYZE/COPY statement keywords it illustrates, and the other control statements it illustrates.

Note that the analyze and copy examples will properly handle tapes with blocksizes up to 256K.

#	EXAMPLE NAME	KEYWORDS	STATEMENTS
1	Analyze any tape		
2	Analyze an SL tape	LENCHK,LBLPRT	PRINT
3	Analyze a partial SL tape	NUMFILES,PRTLEN	PRINT ALL
4	Copy an SL tape	CAT	
5	Copy any tape to an SL tape	VERIFY,KEEP	DROP
6	Partial copy of an SL tape	NUMFILES,REBLOCK, COMPACTION,CAT	RENAME
7	Partial copy of an SL tape in BLP mode		PRINT
8	Analyze a multi-volume tape set		PRINT,SCAN
9	Analyze an SL tape with no label processing	LABELS	
10	Analyze an NL tape	NF,PRTLEN	PRINT,SCAN, TABLE
11	Search a partial file	ENDAFTER,PRTLEN, LBLPRT	REPLACE
12	Copy an ANSI tape		
13	Copy an NL tape	NUMFILES,VERIFY	KEEP
14	Copy NL files to SL	NUMFILES	
15	Image copy a tape	LABELS	
16	Copy a multi-file, multi-volume to 3590	VERIFY,CAT	RENAME
17	Copy and modify a tape file	MSGLEVEL,CAT	REPLACE
18	Copy and modify a tape file	CAT	REPLACE
19	Recover from data checks	RETRY,KEEP	REPLACE
20	Recover from data checks on multi-volume tape		
21	Recover from data checks	LABELS	
22	Recover data from an overwritten tape	NF,CAT	
23	Copy files from an overwritten tape	CAT	RENAME
24	Copy an improperly closed tape	ENDAFTER	
25	Exclude files from a multi-file copy	VERIFY	DROP
26	Copy user labels		
27	Drop an extra tape mark	NUMFILES,CAT	DROP
28	FATAR as a FATS sub-task	VOL,CAT	
29	FATAR as a FATS sub-task	VOL,VOLINCR,MAXVOLN, VERIFY	

26.2 EXAMPLES

EXAMPLE 1: ANALYZE ANY TAPE

Analyze any tape. Since LABEL=(,BLP) is specified on the TAPEIN DD Statement, any tape can be mounted when a mount for volser 999999 is issued; the DSN= in the JCL does not have to be correct. You may need to specify EXPDT=98000 (as shown) so that your tape management system does not attempt to verify volume 999999. The tape may have any type of labels (IBM standard, ANSI, or none at all); FATAR will automatically determine the label type and handle them if present. If the labels are ANSI, both labels and data will be translated to EBCDIC. FATAR will print a summary of the files and data on the tape on the optional TAPESUMM DD Statement. FATAR will print all tape labels, count the blocks in every file, and identify any block causing a temporary or permanent data check, printing the first 80 characters of that block as well as the blocks preceding and following it. All blocks will be deblocked into logical records (if the tape is labeled). All files on the tape will be processed.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=999999,
//            LABEL=(,BLP,EXPDT=98000),DISP=OLD
//SYSIN    DD *
  ANALYZE
```

EXAMPLE 2: ANALYZE AN SL TAPE

Analyze standard label tape 111111. Since it is being opened as a labeled tape (no LABEL parameter on the TAPEIN DD Statement), the volume serial and first data set name must be specified correctly; your tape management system may verify these values. The parameter LENCHK causes FATAR to print the beginning of any block that differs in length from the preceding block (for checking fixed length files). The parameter LBLPRT=DUMP causes labels to be printed in dump format (hex and EBCDIC). The PRINT statement causes blocks 1 to 3 of every data file to be printed. All other operations are the same as Example 1 above.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//            DISP=OLD
//SYSIN    DD *
   ANALYZE LENCHK,LBLPRT=DUMP
   PRINT LF=ALL,B=1-3
```

EXAMPLE 3: ANALYZE A PARTIAL SL TAPE

Analyze the fourth and fifth files on standard labeled tape 111111, where the fourth file is called "FOURTHDS"; your tape management system may verify these values. The LABEL=4 parameter on the TAPEIN DD Statement tells FATAR to begin with that data set, and the NUMFILES=2 parameter on the ANALYZE Statement limits the number of files FATAR will analyze. The PRINT ALL control statement directs FATAR to print all data blocks (first 200 bytes).

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FOURTHDS,UNIT=TAPE,VOL=SER=111111,
//            LABEL=4,DISP=OLD
//SYSIN    DD *
   ANALYZE NUMFILES=2,PRTLEN=200
   PRINT ALL
```

EXAMPLE 4: COPY AN SL TAPE

Standard labeled tape 111111 will be copied to standard labeled tape 222222. All files will be copied. Each file on tape 222222 will receive the data set name of the original file on tape 111111; the data set name on the TAPEOUT DD statement will not be used. If any blocks have unrecoverable data checks, they will be dropped from the output tape, and the block counts in the data set trailer labels adjusted. If you have the FATAR tape management interface enabled, it will be used to get the full 44-character data set names of every input file; if this is successful, CAT=RECAT will cause every output data set to be recataloged to the output tape.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//            DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,VOL=SER=222222,
//            DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY  CAT=RECAT
```

EXAMPLE 5: COPY ANY TAPE TO AN SL TAPE

Tape 999999 will be copied to a scratch standard label volume. All files will be copied. FATAR will determine the label type of the input tape (STANDARD, ANSI, or UNLABELED) and process it appropriately. If the input labels are ANSI, all data files will be translated from ASCII to EBCDIC (if the records are ANSI variable length, RECFM=D, they will be converted to IBM RECFM=V). The data set names from the input tape will be copied to the output (if TAPEIN is unlabeled, the DSN= value from the TAPEIN DD statement is used to name all output files); if the FATAR tape management interface is enabled, it will be used to get the full 44-character data set name of every input file. The volume serial of the output tape will be preserved. If the output tape is filled before the input data is exhausted another volume will be requested. If any blocks on the input have unrecoverable data checks, they will be identified and printed, but the KEEP parameter will cause them to be written to the output tape anyway (using whatever data the input tape was able to transfer). The DROP statements will prevent the indicated input blocks from being written to the output tape. The VERIFY parameter will cause the output tapes to be analyzed after the copy is complete to verify readability.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=999999,
//            LABEL=(,BLP),DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY KEEP,VERIFY
  DROP LF=3,B=2
  DROP LF=3,B=10-12
```

EXAMPLE 6: PARTIAL COPY OF AN SL TAPE

Files 3 and 4 will be copied from labeled tape 111111 and written to a scratch Magstar (3590) tape. The LABEL=3 parameter on the TAPEIN DD Statement causes FATAR to start the copy with file 3 of the tape; the data set name of that file must be properly specified. The NUMFILES=2 parameter tells FATAR to copy two logical files, namely files 3 and 4. The files will become files 1 and 2 on the output tape, with the data set names of the original files. COMPACTION=COMP will cause the output tape to be written using IDRC hardware compaction. REBLOCK=16000 will cause each copied file to be reblocked to as close to a blocksize of 16000 bytes as possible. The RENAME statement is used to specify new names for the 2 output files, and CAT=YES will cause FATAR to catalog those names to the output tape.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=THIRDDS,UNIT=TAPE,VOL=SER=111111,
//            LABEL=3,DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=3590-1,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY NUMFILES=2,REBLOCK=16000,COMPACTION=COMP,CAT=YES
  RENAME LF=3,NEWN=TECHSERV.FILE3
  RENAME LF=4,NEWN=TECHSERV.FILE4
```

EXAMPLE 7: PARTIAL COPY OF AN SL TAPE

All files on labeled tape 111111 from file 4 through the end of the tape are copied to labeled output tape 222222, starting at file 2. In other words, file 4 will be copied to file 2, file 5 will be copied to file 3, etc. Data set names from the input tape will be retained; if the FATAR tape management interface is enabled, it will be used to get the full 44-character data set name of every input file. DCB=RECFM=U on the input tape will cause FATAR to treat the blocks as unblocked for printing/modification purposes but will not affect the DCB that will be written in the output tape's labels (that will be copied from the input tape's labels). LABEL=(10,BLP) is required on TAPEIN since we do not know its data set name and since FATAR must be positioned to the header labels of the first file. The rule is "SL file number, times 3, minus 2", in this case 4*3 – 2 = 10.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=UNKNOWN,UNIT=TAPE,VOL=SER=111111,
//            LABEL=(10,BLP),DISP=OLD,DCB=RECFM=U
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,VOL=SER=222222,
//            LABEL=2,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY
  PRINT ALL
```

EXAMPLE 8: ANALYZE A MULTI-VOLUME TAPE SET

Analyze multi-volume labeled tape set 333333, 444444, and 555555, that contain more than one data set. FATAR will properly read all three volumes, giving total block counts for each data set even when it crosses volumes (an intermediate block count is given each time an end-of-volume is reached). The tape summary (DD statement TAPESUMM) will contain one page for each tape, detailing the files and block counts on it. File numbers will be logical files since the beginning of the first tape, counting files that cross multiple volumes as one file. The PRINT control statements instruct FATAR to PRINT block 7 of file 1 in hexadecimal, blocks 22 and 23 of file 2 in dump format (HEX and EBCDIC) printing 96 bytes, and block 125 of file 4 in EBCDIC printing 250 bytes. The SCAN control statement requests FATAR to verify that the first 4 bytes of every record in file 2 contain a valid zoned decimal number.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=UNKNOWN,UNIT=TAPE,
//            VOL=SER=(333333,444444,555555),
//            LABEL=(,BLP,EXPDT=98000),DISP=OLD
//SYSIN    DD *
  ANALYZE
  PRINT LF=1,B=7,HEX
  PRINT LF=2,B=22-23,L=96,DUMP
  PRINT LF=4,B=125,L=250
  SCAN LF=2,B=ALL,L=4,P=1,T=Z
```

EXAMPLE 9: ANALYZE AN SL TAPE WITH NO LABEL PROCESSING

Analyze SL tape 111111 without label processing, so that labels will be processed as data (however, labels will still be formatted and printed). Processing will continue until two consecutive tape marks are read or until EOV labels are read (tapes containing data sets that overflowed to another tape and end with EOV labels will not end with two tape marks, so FATAR has a special SCAN for the EOV labels). Since labels are not processed and no DCB information appears in the JCL, FATAR will process all tape blocks as RECFM=U (unblocked). Since FATAR is not processing labels, logical file numbers refer to the physical file number relative to the beginning of the tape, and the header and trailer label files are counted.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//            DISP=OLD
//SYSIN    DD *
  ANALYZE LABELS=NO
```

EXAMPLE 10: ANALYZE AN NL TAPE

Analyze unlabeled tape 666666 with deblocking. The NF=3 parameter limits FATAR to reading three physical files. Since DCB information is supplied on the TAPEIN DD statement, FATAR will attempt to deblock every file processed. On the PRINT and SCAN statements, the LF= parameter refers to the physical file number since the beginning of the tape. The SCAN statement is used to VERIFY that bytes 4 to 6 of all records from record 5 in block 3 to record 2 in block 5 contain only characters A, B, or C (as defined in the TABLE statement). The PRTLEN=5000 will ensure that any blocks printed (due to the PRINT statement or failure of the SCAN statement) will print the entire block (up to 5000 bytes), deblocked into records.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=NOLABEL,UNIT=TAPE,VOL=SER=666666,
//            DCB=(RECFM=FB,LRECL=120),
//            LABEL=(,NL),DISP=OLD
//SYSIN    DD *
  ANALYZE NF=3,PRTLEN=5000
  PRINT LF=1,B=1-20
  TABLE T=A,D=C'ABC'
  SCAN LF=2,B=3.5-5.2,L=3,P=4,T=A
```

EXAMPLE 11: SEARCH A PARTIAL FILE

Analyze a partial file on SL tape 111111 and print blocks containing a certain string. Since only the first 20 blocks are to be scanned, there is no need to read the rest of the tape; the ENDAFTER parameter causes FATAR to terminate immediately after processing block 20 in file 1 (since labels are being processed, this is the first data file). The REPLACE statement tells FATAR to print any block containing the string "EQUIPMENT" in any position, and the PRTLEN=32767 parameter causes the entire data block to be printed. The LBLPRT=HEX parameter causes the header labels of the file to be printed in hexadecimal.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=UNKNOWN,UNIT=TAPE,VOL=SER=111111,
//            LABEL=(,BLP,EXPDT=98000),DISP=OLD
//SYSIN    DD *
  ANALYZE ENDAFTER=(LF=1,B=20),PRTLEN=32767,
              LBLPRT=HEX
  REPLACE LF=1,B=1-20,S=C'EQUIPMENT'
```

EXAMPLE 12: COPY AN ANSI TAPE

ANSI labeled tape 333333 is to be copied to a scratch tape with ANSI labels. The TAPEIN DD statement specifies LABEL=(,AL), but it could also specify LABEL=(,BLP) if the true volume serial or first data set name were not known; FATAR works correctly in either case. The TAPEOUT DD statement must specify LABEL=(,AL) for the labels to be written in ANSI format and the data in ASCII character set. All other processing is just the same as for standard label tapes (see example 4).

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=333333,
//            LABEL=(,AL),DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,
//            LABEL=(,AL),DISP=(NEW,KEEP)
//SYSIN    DD *
  ANALYZE
```

EXAMPLE 13: COPY AN NL TAPE

7 files from unlabeled tape 222222 are to be copied to a scratch unlabeled tape. NUMFILES=7 tells FATAR to copy 7 files (7 tape marks and all data preceding them). Note that NUMFILES=0 could have been specified or defaulted to cause FATAR to copy all files until two consecutive tape marks were read. If the FATAR tape management interface is enabled, it will be used to get the true data set name of every input file and use it to name the equivalent output file. The first KEEP statement causes block 7 of file 2 to be written to the output tape even though it is known to cause a data check. The second KEEP statement causes block 12 of file 3 to be increased in length to 1400 bytes, with the original data block positioned at byte 51 (50 bytes of spaces will be added to the start of the block; if the original block is less than 1350 bytes long, additional spaces will be added to the end). The VERIFY parameter will cause the output tape to be analyzed for readability; however, if the output tape requires multiple volumes, the verify will be suppressed (multi-volume NL tapes cannot be verified).

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=222222,
//            LABEL=(,NL),DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,
//            LABEL=(,NL),DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY NUMFILES=7,VERIFY
  KEEP DCK,LF=2,B=7
  KEEP LF=3,B=12,L=1400,P=51
```

EXAMPLE 14: COPY NL FILES TO AN SL TAPE

Two unlabeled files from tape 003168, starting with the third file, are to be copied to a new tape as labeled files. LABEL=(3,NL) on the TAPEIN DD positions the input tape to start with file number 3. LABEL=(,SL) on the TAPEOUT DD statement instructs FATAR to mount a scratch tape and copy the input files to standard labeled output files. FATAR creates labels for each file, and uses the TAPEOUT DSN= parameter to name each of the output files JAT.COPY.TESTNL.DATA. The NUMFILES=2 parameter limits FATAR to copying two files.

```
//FATAR  EXECPGM=FATAR,REGION=0M
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SYSPRIN1  DD SYSOUT=*
//TAPESUMM  DD SYSOUT=*
//TAPEIN    DD DSN=NOT.USED,UNIT=TAPE,VOL=SER=003168,
//             LABEL=(3,NL),DISP=OLD
//TAPEOUT   DD DSN=JAT.COPY.TESTNL.DATA,UNIT=TAPE,
//             LABEL=(,SL),DISP=(NEW,KEEP)
//SYSIN     DD *
 COPY NUMFILES=2,VERIFY
```

EXAMPLE 15: IMAGE COPY OF A TAPE

When a tape is damaged (but readable) or suspect (prone to failure) you can copy the tape to another output volume with an "image copy" of the tape. This involves copying all data from the input to the output tape including all labels and the volume serial, creating an exact bit-for-bit copy. Since the output tape will have the same volume serial as the original, tape management systems and system catalogs will usually not have to be modified or updated. LABEL=(,BLP) is shown on the TAPEOUT DD statement in this example, but even if omitted the OUTBLP option is forced by LABELS=IMAGE so that all labels and data can be copied to the output tape. LABEL=(,BLP) is optional on TAPEIN.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=ANYDSN,UNIT=3490,VOL=SER=111111,
//            LABEL=(,BLP),DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=3490,
//            LABEL=(,BLP),DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY LABELS=IMAGE
```

EXAMPLE 16: COPY A MULTI-FILE, MULTI-VOLUME TAPE TO A 3590 CARTRIDGE

Standard labeled tapes 333333, 444444, and 555555 comprise a multi-volume, multi-file tape set on 3480 cartridges, containing 17 files, and are to be copied to a 3590 Magstar cartridge. The VERIFY parameter causes the resulting 3590 volume to be analyzed for readability. If the FATAR tape management interface is enabled, it is used to get the full data set name of every input file; the RENAME statement with NEWI=.ABC will cause the second level index of every input data set name to be replaced with "ABC" when the files are opened on the output tape and CAT=ONLY causes those output files to be cataloged if they were cataloged to the input tape. For example, if an input data set name was "PAYROLL.PROD.MASTER", the output data set becomes "PAYROLL.ABC.MASTER".

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=3480,
//            VOL=SER=(333333,444444,555555),
//            DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=3590-1,
//            DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY VERIFY,CAT=ONLY
  RENAME  LF=ALL,NEWI=.ABC
```

EXAMPLE 17: COPY AND MODIFY A FILE

A mailing list file on tape has a company name misspelled in many places; we want to copy the file and change the name wherever it appears. The REPLACE statement causes all occurrences of "INVENTION DATA PROCESSING" to be changed to "INNOVATION DATA PROCESSING" whenever it appears in column 17 of any record. Note the way that the character strings are continued; multiple C' or X ' strings following a V=, S= or D= will simply be concatenated (up to 256 bytes). The MSGLEVEL=0 parameter suppresses the printing of the data portion of the modified blocks. CAT=RECAT will cause the file to be recataloged to the output tape.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=COMPANY.MAIL.MASTER,
//            DISP=(OLD,KEEP)
//TAPEOUT  DD DSN=COMPANY.MAIL.MASTER,UNIT=CART,
//            DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY MSGLEVEL=0,CAT=RECAT
  REPLACE LF=1,B=ALL,P=17,V=C'INVENTION DATA',
       C'PROCESSING',D=C'INNOVATION DA',
            C'TA PROC',C'ESSING'
```

EXAMPLE 18: COPY AND MODIFY A FILE

Tape master file "MASTER" on labeled tape 111111 causes a run-time data exception when it is read by its processing program. A previous FATAR run has revealed that record 3 of block 27 contains "00735B4A" in hexadecimal, which is an invalid packed decimal number. It should contain "0073534C". The REPLACE statement causes this substitution to be made. CAT=ONLY will recatalog the data set to the output tape if the input tape file was cataloged.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=MASTER,UNIT=TAPE,VOL=SER=111111,
//            DISP=OLD
//TAPEOUT  DD DSN=MASTER,UNIT=TAPE,
//            DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY CAT=ONLY
  REPLACE LF=1,B=27.3,S=X '00735B4A',
       D=X '0073534C'
```

EXAMPLE 19: RECOVER FROM DATA CHECKS

Unlabeled tape 222222 is known to have data checks in a number of blocks. Copy the tape to another unlabeled scratch, retrying all data checks 256 times in each direction to insure that all possible attempts have been made to recover the data. If any block still cannot be read, the KEEP parameter will cause it to be written to the output tape anyway, but the REPLACE DCK statement will cause the first four bytes of those records to be set to hexadecimal "FFFFFFFF" (high values) that will signal the application program to perform special validation. The DCB parameters are provided on the TAPEIN DD statement so that the input records can be deblocked.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=NLINPUT,UNIT=TAPE,VOL=SER=222222,
//            LABEL=(,NL),DISP=OLD,DCB=(RECFM=FB,LRECL=240)
//TAPEOUT  DD DSN=NLOUTPUT,UNIT=TAPE,
//            LABEL=(,NL),DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY RETRY=256,KEEP
  REPLACE DCK,F=1,B=ALL,P=1,D=X'FFFFFFFF'
```

EXAMPLE 20: RECOVER FROM DATA CHECKS ON MULTI-VOLUME TAPE

Data set "LARGEDS" occupies three tape volumes "AAAAAA", "BBBBBB", and "CCCCCC". Volume "BBBBBB" has developed read data checks, so we want to create a valid copy, dropping the data check blocks. With FATAR, it is not necessary to read the volumes preceding the first data check (in this case "AAAAAA"). The JCL below will produce output volumes that can be used to REPLACE "BBBBBB" and "CCCCCC". However, it is up to the user to update the system catalogs to reflect the new volume serials. If you have a tape management system, you must also do whatever manual updating is necessary to cause its records to properly reflect the new volume relationships. Note that this tape management update will be unnecessary if you copy the entire tape set (as in Example 16) or if you copy only the problem tape (as in Example 21).

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=LARGEDS,UNIT=TAPE,
//            VOL=SER=(BBBBBB,CCCCCC),DISP=OLD
//TAPEOUT  DD DSN=LARGEDS,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY
```

EXAMPLE 21: RECOVER FROM DATA CHECKS

This is another way to recover from the data checks in example 20 above. It does not require any catalog or tape management updates since it creates an exact copy ("image copy") of the bad tape, including its volume serial; in other words, the output of this run is a tape with the same volume serial as the original tape, so it should be externally re-labeled and the original tape discarded. Since tape "BBBBBB" is the middle tape of a multi-volume set, we know that it is completely filled with data; the output tape must be as large or larger than the original so that FATAR can WRITE all of the data and labels before the end-of-tape is detected, or the copy will be invalid. FATAR will correct the trailer label block counts if blocks are dropped because of data checks.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=LARGEDS,UNIT=TAPE,
//            VOL=SER=BBBBBB,DISP=OLD
//TAPEOUT  DD DSN=LARGEDS,UNIT=TAPE,DISP=(NEW,KEEP),
//            LABEL=(,BLP,EXPDT=98000)
//SYSIN    DD *
  COPY LABELS=IMAGE
```

EXAMPLE 22: RECOVER DATA FROM AN OVERWRITTEN CARTRIDGE

A cartridge tape with valid data on it was accidentally used for output, but the new file is smaller than the original file; it is desired to recover as much of the original data file as possible. LABELS=EOD causes FATAR to search for the first hardware EOD mark on the tape and recover data that follows it. Since header labels are not available for the overwritten file, the DCB characteristics of the original data should be provided on the TAPEIN and TAPEOUT DD statements, if known. The TAPEOUT DD statement specifies a labeled scratch tape with the data set name of the original data set. The first data block is likely to cause a permanent data check if it is partially overwritten; if this is the only error it can be ignored. If there are additional original files beyond the overlaid file, FATAR will automatically recover them as well, using the data set name and DCB information in their header labels to copy to the output tape. See Section 24.2 for additional details on overwritten tapes. LABELS=EOD is not supported for round tapes and 3480 cartridges.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=ORIGINAL,UNIT=TAPE,
//            VOL=SER=BBBBBB,DISP=OLD,
//            DCB=(RECFM=FB,LRECL=125,BLKSIZE=32750)
//TAPEOUT  DD DSN=ORIGINAL,UNIT=TAPE,DISP=(NEW,KEEP),
//            DCB=(RECFM=FB,LRECL=125,BLKSIZE=32750)
//SYSIN    DD *
  COPY LABELS=EOD,BLP
```

EXAMPLE 23: RECOVER DATA FROM AN OVERWRITTEN TAPE

A round tape or 3480 cartridge with valid data on it was accidentally used for output, but the new file is smaller than the original file; it is desired to recover as much of the original data file as possible. The new file will occupy four physical files on the tape (header labels, data, trailer labels, and extra tape mark), so the LABEL=(5,BLP) will position FATAR immediately beyond the extra tape mark, at the beginning of the remaining original data. FATAR will process this as unlabeled data since it is not positioned to label records. Since labels are not available, the DCB characteristics of the original data should be provided on the TAPEIN and TAPEOUT DD statements, if known. The TAPEOUT DD statement specifies a labeled scratch tape with the data set name of the original data set. The first data block is likely to cause a permanent data check if it is partially overwritten; if this is the only error it can be ignored. NF=1 causes FATAR to read only the original data file. CAT=RECAT will cause the data set to be recataloged to the output tape (be sure to specify the correct data set name on the TAPEIN DD statement). See Section 24.2 for additional details on overwritten tapes.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=ORIGINAL,UNIT=TAPE,
//            VOL=SER=BBBBBB,LABEL=(5,BLP),DISP=OLD,
//            DCB=(RECFM=FB,LRECL=125,BLKSIZE=32750)
//TAPEOUT  DD DSN=ORIGINAL,UNIT=TAPE,DISP=(NEW,KEEP),
//            DCB=(RECFM=FB,LRECL=125,BLKSIZE=32750)
//SYSIN    DD *
  COPY NF=1,CAT=RECAT
```

EXAMPLE 24: COPY FILES FROM AN OVERWRITTEN TAPE

A round tape or 3480 cartridge with multiple files on it was accidentally used for output, but the new file is smaller than the original first file; it is desired to copy the original files beyond the first. The new file will occupy four physical files on the tape (header labels, data, trailer labels, and extra tape mark) and the original first file will occupy two more (data file and trailer labels) so the LABEL=(7,BLP) will position FATAR to the header labels of the original second file. FATAR will process this as a labeled tape; DCB information will be copied from the labels of the copied files. FATAR can copy data set names from the original tape labels, but these data sets will no longer be recorded in your tape management system so FATAR will not be able to get the full original data set names; if the original names are longer than 16 characters, the RENAME statements can be used to specify the correct names; CAT=RECAT will cause them to be recataloged to the new output tape. See Section 24.2 for additional details on overwritten tapes.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=ORIGINAL,UNIT=TAPE,
//            VOL=SER=BBBBBB,LABEL=(7,BLP),DISP=OLD
//TAPEOUT  DD DSN=NOTUSED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY  CAT=RECAT
  RENAME  LF=3,NEWN=ORIGINAL3
  RENAME  LF=4,NEWN=ORIGINAL4
```

EXAMPLE 25: COPY AN IMPROPERLY CLOSED TAPE

Due to a system crash or power failure, tape 333333 was not properly closed and has no trailing tape mark or labels; it cannot be read with standard access methods. A previous FATAR analyze job shows that block 742 is the last good block of the proper length; a data check occurs immediately after it. The job below will copy the tape up to and including block 742 onto a scratch tape with proper labels.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=A.NAME,UNIT=TAPE,
//            VOL=SER=333333,DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY ENDAFTER=(LF=1,B=742)
```

EXAMPLE 26: EXCLUDE FILES FROM A MULTI-FILE COPY

Multi-file tape 444444 is to be copied to a scratch tape, but files 3 and 5 are not to be copied. The DROP statements with B=ALL will cause those files, including all of their labels and tape marks to be entirely omitted from the output tape. This means that file 4 from the input tape becomes file 3 on the output, file 6 from the input becomes file 4, file 7 comes file 5, etc. The VERIFY parameter causes the output tape to be read back to verify readability, number of files, labels, etc.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,
//            VOL=SER=444444,DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY VERIFY
  DROP LF=3,B=ALL
  DROP LF=5,B=ALL
```

EXAMPLE 27: DROP AN EXTRA TAPE MARK

Because of a malfunctioning tape drive, labeled tape 555555 has an extra tape mark following the trailer labels for file 3. This makes all files past 3 unusable since data management expects an exact sequence of labels and tape marks; FATAR also cannot process the whole tape as labeled, but is capable of processing part of it. Two techniques are shown for recovering from this error:

The first technique causes an exact copy of the tape to be created, but without the extra tape mark. Since the tape mark looks like a file with no data blocks, the DROP statement shown will omit that file and therefore drop the tape mark (it is file 10 since the three standard label files preceding it appear to be three files each in LABELS=NO mode). The output tape will have the serial number of the original tape and should be externally re-labeled.

```
//FATAR    EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,
//            VOL=SER=555555,DISP=OLD
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,
//            LABEL=(,BLP),DISP=(NEW,KEEP)
//SYSIN    DD *
  COPY LABELS=IMAGE
  DROP LF=10,B=ALL
```

The second technique involves copying the data up to and beyond the extra tape mark in two separate FATAR steps, but processing in label mode. The first step copies the first three good files. LABEL=(11,BLP) in the second step positions the input tape to the header labels of the file beyond the tape mark, and LABEL=4 on the output tape causes FATAR to start the copy in the right place. The second step then continues to the end of the tape. The output tape will retain its original serial number, and labels will be properly updated if needed. If the FATAR tape management interface is enabled, it will be used to get the full data set name of each input data set; CAT=RECAT will cause them to be cataloged to the output tape.

```
//FATAR1   EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=555555,
//            LABEL=(1,BLP),DISP=(OLD,PASS)
//TAPEOUT  DD DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,PASS)
//SYSIN    DD *
  COPY NUMFILES=3,CAT=RECAT
/*
//FATAR2   EXEC PGM=FATAR,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPEIN   DD DSN=FIRSTDS,UNIT=TAPE,VOL=SER=555555,
//            LABEL=(11,BLP),DISP=(OLD,KEEP)
//TAPEOUT  DD DSN=NOT.USED,DISP=(NEW,KEEP),
//            VOL=REF=*.FATAR1.TAPEOUT,LABEL=4
//SYSIN    DD *
  COPY CAT=RECAT
```

EXAMPLE 28: FATAR AS A FATS SUB-TASK COPY TO MAGSTAR 3590

FATAR may be run as a sub-task of FATS, and FATS parameters may be used to call for multiple tape volumes on the same drive. However, each tape is treated as a single volume, not as a multi-volume set. FATAR will be re-initialized for each tape processed. LABEL=(,BLP) is used on the TAPE1 DD statement since the tapes will probably have different data set names, and "DDDDDD" is a dummy volume serial that will not be used. The VOL= parameter will causes tapes 111111, 222222, 333333, and 444444 to be copied. A console mount message will be issued for each tape requested. A 3590 scratch output tape will be mounted for each input tape, and all files copied to it. A full FATAR printout will be printed for each tape, and the FATS summary will show the results for all tapes. If the FATAR tape management interface is enabled, it will be used to get the full data set name of each input data set; CAT=RECAT will cause them to be cataloged to the output tape.

```
//FATFTR   EXEC PGM=FATS
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=ANYDSN,UNIT=(TAPE,,DEFER),
//            VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP,EXPDT=98000)
//TAPE1OUT DD DSN=ANYDSN,UNIT=(3590-1,,DEFER),
//            DISP=(,KEEP)
//SYSIN    DD *
  ANALYZE(1) VOL=(111111,222222,333333,444444),CAT=RECAT
```

EXAMPLE 29: FATAR AS A FATS SUB-TASK MULTIPLE TAPES AND DRIVES

When FATAR is run as a sub-task of FATS, FATS may invoke multiple copies of FATAR to perform operations on separate tape drives. Each FATAR execution will produce a separate report on its own SYSPRINx DD statement. Multiple volumes may be requested on each drive. The example below will request volumes ABC101, ABC102, and ABC103 on TAPE1, and volumes 000123 and 000125 on TAPE2 (copying the files on them to a scratch tape on TAPE2OUT).

```
//FATFTR   EXEC PGM=FATS
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SYSPRIN2 DD SYSOUT=*
//SUMMPRT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1    DD DSN=ANYDSN1,UNIT=(TAPE,,DEFER),
//            VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//TAPE2    DD DSN=ANYDSN2,UNIT=(TAPE,,DEFER),
//            VOL=SER=EEEEEE,DISP=OLD,LABEL=(,BLP)
//TAPE2OUT DD DSN=NOT.USED,UNIT=TAPE,DISP=(,KEEP)
//SYSIN    DD *
  ANALYZE(1) VOL=ABC101,VOLINCR=1,MAXVOLN=3
  ANALYZE(2) VOL=(000123,000125),VERIFY
```

This page intentionally left blank.

30.0 FATSCOPY FUNCTIONAL DESCRIPTION

WHAT IS FATSCOPY?

FATSCOPY uses the copying facilities of FATAR to automate stacking of data sets on high-capacity tape volumes and migration to new types of tape. Only data sets with IBM standard labels (LABEL=SL) or standard user labels (LABEL=SUL) can be copied.

FATSCOPY can select the data sets to be copied or migrated from your system catalogs or from your tape management database. You specify the data sets by:

- data set name prefixes or masks (such as JAT.DATA.**);
- tape volume serial prefixes or ranges;
- catalog tape device type (3480, 3490E or 3590).

Once the data sets are selected from the catalog, FATSCOPY will access information for each data set from your tape management system. FATSCOPY currently supports CA 1, TLMS, and DFSMSrmm; see Section 90.3 for information on installing the required tape management interface. You can then additionally filter the data sets to be copied based on tape management system information, including:

- creation date
- expiration date
- creating job name and/or step name
- data set size

FATSCOPY can also be directed to copy entire tape volumes, selected by volser. The data set names on the specified tapes will be acquired from your tape management system, and all files on those tapes will be copied. If a tape volume specified is part of a multi-volume tape set, FATSCOPY will copy all volumes in that tape set and all data sets on the tape set, automatically.

FATSCOPY will dynamically allocate the input data sets it has selected and will then copy them to an output tape. Information recorded in your tape management system about each input data set can optionally be propagated to the tape management records of the output data sets, and the input tapes can optionally be returned to scratch status for reuse.

A short demonstration of FATSCOPY's features and capabilities is available online at

www.fdr.com/demo.cfm .

FATSCOPY TAPE STACKING

Like all computer technology, tape technology keeps improving. Every few years a new generation of tape drives and tape media allows you to put more data in a tape cartridge with the same physical size. In the 1980s, 3480 tape cartridges held 200MB of data. Tape capacity today using cartridges of the same physical size is close to 5 TB; with compression, over 15 TB can be stored on a single tape.

Unfortunately, many tape data sets are small or vary in size from one run to another, and most tape applications are designed to place only one data set on a tape volume. The result is that many tapes in a tape library contain only a fraction of the data they are capable of holding. This results in inefficient use of the physical space in the library (a potentially expensive problem if an automated tape library (ATL/silo) is involved) and requires more cartridges, an additional expense.

These problems can be minimized and the resulting expenses avoided by putting more data on each cartridge. One way to do so is by placing multiple data sets on each cartridge, also called "stacking" data sets. With the new high-capacity tape cartridges, a large number of data sets can often be stacked on a single tape.

Although it is possible to automate data set stacking by changing application JCL, it can be labor-intensive and error-prone to make the required changes and to monitor and correct them as application needs change. Also, it is subject to problems; for example, if one of the data sets to be stacked on tape is not created, perhaps due to an application ABEND, all the data sets to be stacked following that failed data set will also fail.

FATSCOPY avoids these problems and automates tape stacking by processing data sets that have already been created on tape. Your applications run without modification, creating data sets on tape as they always have. But periodically you can run FATSCOPY to collect all of those inefficient data sets and stack them as consecutive files on fewer tape volumes, releasing the inefficient input volumes as scratch tapes for reuse. The stacked data sets will be re-cataloged, so if they are needed as input they will be mounted and read transparently; the applications will not be aware that the stacking was done.

If input tapes were selected by volser instead of by cataloged data set name, FATSCOPY will not stack tapes by default. All files on a given input tape or tape set will be copied to a single output tape (or tape set), but each new input tape will cause a new output tape to be used. However, you can request that these input files be stacked on the output tapes.

FATSCOPY MIGRATION

FATSCOPY can also be used when you are migrating from one type of tape device to another, such as when you install a new generation of faster, higher-capacity tape drives. Although much of your tape data will automatically migrate to the new tapes as new data sets are created on the new drives and old data sets expire, there will always be some unexpired data sets on the old tapes.

FATSCOPY can be used to copy this residual data to the new tapes, optionally stacking the data sets as well, allowing you to eventually remove the old technology drives.

COPYING FILES RECORDED IN EXTERNAL DATABASES

The locations of certain types of data sets are recorded in external databases separate from tape management systems. In some cases, such as ABR Archive and Application backup data sets and HSM ML2 data sets, FATSCOPY is able to update those databases so that the copied files can be found by those applications.

However, in other cases FATSCOPY does not update the databases. Innovation does not recommend using FATSCOPY to copy files controlled by any other application that maintains information about the location of tape files in an external database.

Please see “Special Tapes” in the FATSCOPY Special Considerations section on page 32-31 for further details on copying these data sets.

If you have any questions about using FATSCOPY with special data set types, contact Innovation for assistance.

30.1 FATSCOPY SECURITY

PROTECTING FATSCOPY EXECUTION AND OPTIONS

FATSCOPY allows you to use your security system to control which users are authorized to execute FATSCOPY. **By default, the only resource that FATSCOPY checks for is FATSCOPY.DATASET.SECBYPAS. Security checking for the other resources is disabled by default; see Section 90 for instructions on enabling those security checks.**

Use of FATSCOPY is controlled by resources defined in the RACF FACILITY class (or its equivalent in other security systems). Consult your IBM RACF or other security vendor documentation for the procedure for defining such FACILITY class resources. Once defined, you must grant READ authority to the resource for users authorized to use it.

For compatibility with earlier releases of FATSCOPY, the default is to allow all users to execute FATSCOPY and use all options. If your security system tells FATSCOPY that a given resource name is not defined, all users will be able to use that function. If you define some but not all of the resources, only those defined functions are restricted. For example, if you don't define FATSCOPY.COPY, all users can use FATSCOPY to copy tapes.

However, some security systems, such as ACF2, protect all resources by default. If these security checks are enabled in such systems, no one will be able to use these FATSCOPY functions until they are defined.

The FACILITY class resources that may be checked by FATSCOPY are:

FATSCOPY.COPY.....provide a TAPEOUT DD statement (copy a tape)

FATSCOPY.DATASET.SECBYPAS..bypass data set security checking for input files

As each control statement is processed, FATSCOPY will issue a RACROUTE to verify that the user has at least READ authority to the proper resource name. The step will fail if RACROUTE returns a code of 8 (or higher) meaning that the resource is defined to the security system but the user associated with the FATSCOPY step does not have authority to it.

In addition, if you are using RMM tape management, you also need to define two additional resources:

STGADMIN.EDG.MASTER ……CONTROL access

STGADMIN.EDG.FORCE ……UPDATE access

INPUT DATA SET SECURITY CHECKING

As FATSCOPY opens each input tape, normal OPEN security checking is done, based on the parameters on your TAPEIN DD statement.

FATSCOPY will first check the FACILITY class resource FATSCOPY.DATASET.SECBYPAS. If this resource is defined, and this user has at least READ authority to it, no DATASET checks will be done in this step. For compatibility with earlier releases of FATSCOPY, if the resource is not defined to your security system, no DATASET checks will be done for any user.

FATSCOPY will issue a RACROUTE for the data set name in the DATASET security class to verify that the user has at least READ authority to the data set name of each input data set.

Note that even if the RACROUTE check is skipped, z/OS OPEN routines issue their own checks, and may fail if the user does not have at least read authority for the data set.

Note: **The input data set security checks are done by default; see Section 90.4 for instructions on disabling then.** In security systems that protect all resources by default, such as ACF2, FATSCOPY.DATASET.SECBYPAS will appear to be protected, so data set security checks will be done for all users by default. If you don't want this, you must either disable the data set security check or define the FATSCOPY.DATASET.SECBYPAS resource.

OUTPUT DATA SET SECURITY CHECKING

FATSCOPY does a normal OPEN on every output data set on TAPEOUT, which will cause standard OPEN security checking to be done by the operating system. The user must be authorized to create a data set with the name specified by FATSCOPY.

31.0 FATSCOPY EXECUTION JCL

To execute FATSCOPY, the following JCL statements are used.

JOB/STEP STATEMENT

If you expect your FATSCOPY job or job step to be processing a large amount of data, you may wish to add TIME=1440 or TIME=NOLIMIT to the JOB or STEP statement to avoid an S322 abend.

EXEC STATEMENT

The EXEC statement specifies the FATSCOPY program name and memory requirements (if your installation defaults are insufficient).

A region of 0M is recommended for FATSCOPY; the program will use no more memory than is required for a particular function.

Example:

```
//FATSCOPY EXEC  PGM=FATSCOPY,REGION=0M
```

STEPLIB/JOBLIB DD STATEMENT

A STEPLIB or JOBLIB DD statement will be required if FATSCOPY has been linkedited into a private library. It can be omitted if FATSCOPY is in a system library that can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). This **must** be an APF authorized library.

If ABRARC=YES is specified to copy ABR Archive and Application Backup data sets, the library that contains the ABR load modules must be added to the STEPLIB/JOBLIB, unless the ABR modules are in a link list library. ABR level 5.4/70 or higher is required to copy ABR tapes with FATSCOPY.

If you are selecting physical tapes on a StorageTek VSM virtual tape system with the PHYSVOL or ALLPHYS keyword, a STEPLIB or JOBLIB DD statement that specifies the link library (SLSLINK) that contains the VTCS and HSC modules is required (unless this library is in the linklist).

SYSPRINT DD STATEMENT

The required SYSPRINT DD statement indicates what data set receives FATSCOPY messages, and is normally allocated to a SYSOUT data set. Its DCB attributes are RECFM=FBA,LRECL=121. If BLKSIZE= is specified it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.

SELRPT DD STATEMENT

SELRPT is a **required** DD statement that receives a report showing all data sets selected from the system catalogs (based on the SELECT/EXCLUDE statements present). If CATDSN is specified, it will be in catalog order. If ALLDSN is specified, it will be in the order in which the data sets are on the volume. It will also show tape management information for each data set, and will indicate in the STATUS and REASON columns whether the data set was actually selected for copying. Some reasons for bypassing a data set include:

- **NOTMINFO –** No tape management information for volume.
- **INV DSN –** Invalid data set name.
- **MISMATCH –** The data set name and location recorded in tape management does not match the name and location recorded in the system catalog.
- **MULTIFIL –** The data set was on a multifile volume and MULTIFILE=NO was used.
- **EXCLUDED –** The data set matched one of the criteria on an EXCLUDE statement.
- **EDM –** The data set is controlled by an external data manager.
- **DUPLICAT –** The data set was already selected by a previous SELECT statement.

SELRPT is usually a SYSOUT data set and its DCB attributes are RECFM=FBA, LRECL=133. If BLKSIZE if specified, it must be a multiple of 133 otherwise it will default to 133 for SYSOUT and 1330 for other devices.

COPYRPT DD STATEMENT

COPYRPT is a **required** DD statement that receives a report showing all data sets actually copied to output tapes, in the order they were copied. It will display the output volser and file sequence number of each output file. It is usually a SYSOUT data set and its DCB attributes are the same as for SELRPT.

COPYRPT2 DD STATEMENT

The optional COPYRPT2 DD statement will receive the same report as COPYPRT, except that there are no report headers or page skips. This is suitable for importation into a spreadsheet or other reporting program. Its DCB attributes are the same as for SELRPT.

TAPESUMM DD STATEMENT

TAPESUMM is **required**. It receives a summary report of all input files in the order they were read, including data set name, tape label information, block and byte counts, and minimum, maximum and average block sizes. It is usually a SYSOUT data set and its DCB attributes are the same as for SYSPRINT.

ERRORRPT DD STATEMENT

The optional ERRORRPT DD statement indicates what data set receives all error messages. It is usually a SYSOUT data set and its DCB attributes are the same as for SELRPT.

AUDIT DD STATEMENT

The AUDIT DD statement is optional. If it is present, FATSCOPY will generate audit records for each data set it copies and write those records to this data set. The audit file must be a disk or tape data set with DCB attributes of RECFM=FB and LRECL=220. Refer to the heading “Using the Audit Data Set” in Section 32.6 for suggestions on allocating and using the audit file. The FATAUDIT program can be used to produce formatted reports using the audit data set, See Section 35 for a description of how to use FATAUDIT.

If an AUDIT DD statement is used, the AUDITLOG= keyword is ignored.

DSNTABLE DD STATEMENT

The DSNTABLE DD statement defines a data set that is used by FATSCOPY to save the selection results of a SIM job or an interrupted COPY job.

- A SIM job with the CHECKPT keyword creates a data set containing the files selected for copying.
- A COPY job that is interrupted by a STOP or CANCEL command creates a data set containing the files that have been selected *and* indicates which of those files have actually been copied.

To create a new DSNTABLE data set, use DISP=(,CATLG). If you code a DCB parameter, use RECFM=FB, LRECL=334, and BLKSIZE=16700. (Note: these values are different from previous FATSCOPY versions.) If you omit the DCB parameter, FATSCOPY will use the correct values by default.

A DSNTABLE data set is also used by a subsequent COPY job with the RESTART keyword to select which files should be copied.

- When using a DSNTABLE created by a SIM job, the COPY RESTART job copies all the files selected by that SIM job.
- When using a DSNTABLE created by an interrupted COPY job, the COPY RESTART job copies the files that were selected but not yet copied by the interrupted job.

To use an existing DSNTABLE data set in a RESTART job, use DISP=SHR.

SYSABEND DD STATEMENT

SYSABEND requests an abend dump if major errors occur. Note that most internal abends in FATSCOPY are for the user's information only and do not normally cause dumps. SYSABEND is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by Innovation. If you have the ABEND-AID product from COMPUWARE, include the following so that a fully-formatted dump is produced:

```
//ABNLIGNR  DD  DUMMY
```

TAPEOUT DD STATEMENT

The TAPEOUT DD statement specifies the output tape onto which FATSCOPY will copy the requested data sets. The TAPEOUT DD must include DSN= and UNIT=. VOL=, DISP=, and RETPD= are optional (except that DISP= is required when your tape management is TLMS), but should be coded using the following guidelines.

DSN= A valid permanent data set name, to meet z/OS requirements. However, this name is treated by FATSCOPY as a dummy name. Actual output data set names will be copied from the input tapes. (**To run concurrent FATSCOPY jobs, use different dummy data set names for each job.**)

Note: If you have SMS routines that determine device allocation using data set names, then you must select a dummy data set name that will allow the proper output device to be allocated to your copy job.

UNIT= A unit name that will allocate the desired type of output tape.

If you are writing data sets to VTFM volumes, use UNIT=(unitvalue,,DEFER). This will guarantee that the VDB entries created by VTFM will contain the first data set name created by FATSCOPY, instead of the data set name coded in the DSN= parameter on the TAPEOUT DD.

VOL= Optionally specify the volume serial of the output tape. If omitted, z/OS will request a scratch tape. Since the default maximum number of volumes that z/OS will allow for an output data set is 5, **Innovation strongly recommends coding VOL=(,,,255) to avoid S837 abends in the event that more than 5 output tapes are required.**

DISP=(NEW,KEEP) **This is required if you are using TLMS tape management**. Do not specify CATLG since FATSCOPY will handle cataloging of the output data sets internally. Change NEW to OLD if you are adding files to a non-scratch volume (using the OUTFSEQ keyword, or LABEL=n on TAPEOUT).

RETPD= or EXPDT= Specify one of these **only** to assign a specific expiration date to **all** of the output data sets. Do not use EXPDT=98000 if you are using the IMAGE keyword.

For a SIM (Simulate) operation, the TAPEOUT DD statement may be omitted. If a TAPEOUT DD statement is present for a SIM and RETPD or EXPDT is specified, FATSCOPY will use the RETPD or EXPDT value to calculate the expiration dates for the output data sets that will be displayed in the SIM Report. Otherwise, the TAPEOUT DD statement is ignored for a SIM.

If you omit RETPD= or EXPDT= from the TAPEOUT DD, FATSCOPY will assign the expiration date of each output file using the expiration date of each input file from tape management. If EXPIRED=YES is specified (to copy expired data sets), and the input data set is expired, the expiration of the output data set is set to the current date plus 2 days.

If you specify STORCLAS= on the TAPEOUT DD, or you specify a DSN= that is assigned a SMS storage class by your SMS ACS routines, the output tape may be SMS-managed. However, FATSCOPY will not copy SMS class information from the input data sets to the output data sets, so the SMS classes specified or assigned will be used for all output data sets.

For example,

```
//TAPEOUT  DD  DSN=ANYNAME,UNIT=CART,DISP=(NEW,KEEP)
//TAPEOUT DD DSN=FILE.NAME,UNIT=3590-1,
//    VOL=SER=444444,DISP=(,KEEP)
```

You can optionally specify a file sequence number in the LABEL= parm on the TAPEOUT DD to tell FATSCOPY to add data sets to a volume that already contains stacked data sets; however, the OUTFSEQ= keyword is the preferred method to use.

ARCHIVE DD STATEMENT

(ABR customers only) The ARCHIVE DD statement is used to specify the name of the ARCHIVE Control File that will be updated if any ABR Archive or Application Backup data sets are copied due to the use of ABRARC=YES. If ABRARC=YES is specified and no ARCHIVE DD is present in the JCL, FATSCOPY will attempt to dynamically allocate the Archive Control File using the Archive DSN in the FDR Options Table.

MAPTAPE DD STATEMENT

The MAPTAPE DD statement is required when using the PHYSVOL or ALLPHYS keywords with an IBM TS7700 virtual tape system. It describes a standard label tape data set **on the virtual tape device containing the input volumes to be copied**. This file is used by FATSCOPY to request tape mapping information from the VTS, and by the VTS to communicate the results back to FATSCOPY. Use the DCB characteristics from the following example:

```
//MAPTAPE  DD UNIT=IBMVTS1,DISP=(,CATLG),DSN=JAT.MAPTAPE.FILE,
//            LABEL=(,SL),RETPD=1,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,TRTCH=NOCOMP)
```

This data set is used by FATSCOPY for both SIM and COPY jobs. If you are using the ISPF panels to do an online SIM with the PHYSVOL or ALLPHYS keyword, your TSO userid must have the MOUNT attribute.

SYSIN DD STATEMENT

The **required** SYSIN DD statement is the source of FATSCOPY control statements. It is normally a "DD *" spool file, but can be any data set with DCB characteristics RECFM=FB and LRECL=80.

This page intentionally left blank.

32.0 FATSCOPY CONTROL STATEMENTS

32.1 GENERAL

GENERAL RULES

All FATSCOPY control statements must be input on 80-character records, of which only columns 1 to 71 are used by FATSCOPY.

Every FATSCOPY control statement must begin with an operation keyword from the list below. This keyword may begin in column 1 or be preceded by any number of spaces. It must be followed by one or more spaces. Every operation keyword accepts a number of operands that follow those blanks and are separated by commas with no intervening spaces. Control statements may be continued onto multiple input records by following the last operand with a comma and a space, and starting the next operand on the next record in any column (preceded by spaces).

Any FATSCOPY control statement may contain comments; FATSCOPY stops its scan of a control statement whenever it finds an operand followed by a space (or comma-space for continuation), so the rest of the statement may be filled with any comment desired. Any control statement record containing an asterisk ("*****") in column 1 will be bypassed by FATSCOPY and may entirely contain comments.

The FATSCOPY operation keywords are:

COPY	Initiates the FATSCOPY operation and specifies general options
SIMULATE	Initiates a FATSCOPY simulation and specifies general options
SELECT	Specifies the data sets or tape volumes to be copied
EXCLUDE	Excludes certain data sets that would otherwise be selected by the SELECT statements
RENAME	Renames data sets during the copy

32.2 COPY/SIM STATEMENT

COPY
SIMULATE
SIM

,ABEND
,ABRARC=NO|YES
,ABRINDEX=prefix
,ADDXDAYS=nnn|4
,ARCBACKUP=DSF
,ARCB1DSN=Dsname
,AUDITLOG=logstreamname
,BUFNO=nn|10
,CAT=ONLY|NO|YES|RECAT
,CHECKPT
,COPYOWNR
,DUMP=NO|ASK|YES
,ECHO=NO|YES
,EXCRPGM=(pgmname,…,pgmname)
,EXLASTJOB=(jobname,…,jobname)
,EXLASTPGM=(pgmname,…,pgmname)
,EXPDTGROUP=nnn|7|999
,EXPIRED=YES|NO
,GROUPBY=VOLSET|EXPDATE
,HSMBAKMASK=hsmmask
,HSMMIGMASK=hsmmask
,HSMML2=NO|YES
,IMAGE
,LASTFILE=nnnn|1000 *
,LASTAPE=dsname
,LASTMULT
,LINECNT=nnnnn|60
,LOGERROR=ABEND|CONTINUE
,MAJERR=nnnnn|1
,MAXERR=nnnnn|1
,MAXFILE=nnnnn|0
,MAXRBLK=nnnnnn
,MAXTERR=nnnnn|100
,MAXTOTFILE=nnnnn|1000 *
,MULTIFILE=NO|YES *
,MULTIVOL=YES|NO|ONLY
,NOSUPULAB
,NOTCATRC=0|4|8
,OFFSITE=NO|ONLY|YES
,OPCAP=nnnn|10
,OUTFSEQ=nnnnn|1
,REBLOCK=nnnnnn
,RESTART
,RETCODE
,RETRY=nnn|0
,RMMFAIL=CONTINUE|STOP|RETRY
,SELTERR=YES|NO
,SHOWNEWN
,SORT=YES|NO
,SUPULAB
,THRESHOLD=nn
,TMSDATA=COPY|NOCOPY
,TMSIN=RMM|CA1|TLMS
,TMSINPUT=KEEP|SCRATCH
,TMSOUT=RMM|CA1
,UNITIN=unitname
,VIRTTYPE=IBM|STK
,VIRTUNIT=esoteric

* these operands are ignored if SELECT ALLDSN statements are used to select tape volumes to copy.

Note: the underlined values are the default values as delivered by Innovation. The values seen by an application programmer may have been changed by your system programmer when installing FATSCOPY, as described in “Operands” on page 32-4.

A COPY or SIM control statement is required and **must be the first statement.** However, all of its operands are optional. Operands may appear in any order on the statement.

A SIMULATE (SIM) statement causes FATSCOPY to simulate the copy operation. It will scan the catalogs and tape management records according to your SELECT/EXCLUDE statements and it will display the data sets that it will select and show the order that they will be copied to output tapes. SIM will not open TAPEOUT (which can be omitted) and it will not mount or read any input tapes. By default, SIM will create an estimate of the number of output tapes needed to copy all of the data sets and a report showing the VOLSERs of the input tapes needed for the job. SIM can be used in conjunction with the CHECKPT keyword to create a data set to be used as input to a subsequent FATSCOPY job to actually copy the data sets.

OPERANDS The following operands can be specified on a COPY/SIM statement. For each of the operands below, the default value set by the FATSCOPY program or contained in the FATSCOPY Option Table shipped with the FATSCOPY product follows the operand. The values for many operands can be changed using the FATZAPOP program (described in Section 90.4.1). **Any of the operands whose defaults are followed by an asterisk (*) may be changed by the system programmer at your installation.** To display the options currently in effect in your installation, run the FATZAPOP program with the PRINT control statement. You should use the ECHO=YES operand on the FATSCOPY COPY/SIM statement if you have any question about the options currently in effect in your installation.

ABEND ABEND causes FATSCOPY to terminate with a U0888 or U0300 abend rather than a return code (condition code) of 12 when serious errors are found. ABEND cannot be used when RETCODE is also specified.

Note: When ECHO=YES has been used and ABEND has been specified or used by default, this value will be shown in the ECHO output as

ABEND ABNORMAL TERMINATION OPTION----------U0888

Delivered default(*): ABEND

ABRARC= (ABR customers only) Controls whether ABR Archive or Application Backup data sets can be selected for copying:

YES - ABR Archive and Application Backup data sets that match the data set selection criteria will be automatically included in the files that are copied. FATSCOPY will update the ABR Archive Control File or ABR Application Control File to reflect the new volume(s) containing the archive and application backup data sets that have been copied.

If the ARCHIVE DD JCL statement has been used, then the data set specified by that DD statement is assumed to be the ABR Control file to be updated. If no ARCHIVE DD is present, the FDR Options Table will be used to determine the name of the Archive Control File to update.

If the ABR modules are not in a LINKLIST library, a STEPLIB or JOBLIB DD must be included for the ABR load library. ABR level 5.4/70 or higher is required.

NO – ABR Archive and Application Backup data sets will ***not*** be selected, even when they otherwise match the data set selection criteria.

Delivered default(*): ABRARC=NO

ABRINDEX= (ABR customers only) Specifies the ABR prefix (first index level) of the ABR ARCHIVE and BACKUP files to be copied. This may be used with FDR Application backups.

Delivered default(*): The ABRINDEX value in the FDR Global Options Table, usually “FDRABR”

ADDXDAYS= If TMSINPUT=SCRATCH is specified, this value (0 to 999) is added to the current date to calculate the date on which input tapes will be scratched. Although **Innovation strongly recommends using the default value,** RMM-controlled tapes with Vital Record Retentions will be retained indefinitely unless ADDXDAYS=0 is specified.

Delivered default(*): ADDXDAYS=4 (retain for 4 days, then scratch)

ARCBACKUP=DSF (ABR customers only) When copying Archive or Application Backup files, this parameter directs FATSCOPY to invoke FDRDSF to DUMP the ABR Control File used in this step as the last file on the tape once all files have been copied. This operates the same as the ARCBACKUP=DSF option of ABR. If no ABR archive or backup data sets were copied, FATSCOPY will not backup the ABR Control Files.

The file used as the ABR Control File is that specified on the ARCHIVE DD JCL statement, if present; otherwise, the FDR Options Table will be used to determine the name of the ABR Control File to update. If the ABR modules are not in a LINKLIST library, a STEPLIB or JOBLIB DD must be included for the ABR load library.

ARCB1DSN= (ABR customers only) Specifies the data set name to be used on the output tape for the backup of the Archive Control File that is done if the ARCBACKUP=DSF operand has also been specified.

Delivered default: The index level "ARCHIVE" in the existing ABR Control File data set name will be changed to "ARCBKUP".

AUDITLOG= "logstreamname" specifies the name of a system logger log stream which is being used to record FATSCOPY audit records. When AUDITLOG= is used, an audit record of each data set copied is written to the system logger. See "Writing Audit Records" on page 32-35 for information on using the system logger to write audit records.

If an AUDIT DD statement is also used, the AUDITLOG= keyword is ignored, and all audit records will be written to the sequential data set specified by the AUDIT DD statement.

BUFNO= Specifies the number of buffers FATSCOPY will use to copy data. Values are from 2 to 25.

Delivered default(*): BUFNO=10

CAT= Controls cataloging of data sets copied to the output tape:

ONLY – output data sets will be cataloged only if the equivalent input data sets are cataloged. The catalog must point to the matching volser and file sequence number of the input data set.

NO - output data sets will not be cataloged.

YES - output data sets will be cataloged if they are not already cataloged.

RECAT - output data sets will be cataloged; if they are already cataloged, the catalog will be updated.

Delivered default(*): CAT=ONLY

CHECKPT (SIM only) Used to create a data set that can be used as input to a subsequent FATSCOPY job that will copy all of the data sets selected by the simulation. The CHECKPT keyword requires a DSNTABLE DD in the JCL.

COPYOWNR (RMM only) Used to specify that the RMM Owner information for the output volume will be copied from the Owner information for the input volume when SELECT CATDSN is being used. (By default, the owner information is propagated only when SELECT ALLDSN is used.) If TMSDATA=NOCOPY is also specified, then COPYOWNR is ignored.

Note: If COPYOWNR is specified and FATSCOPY stacks multiple volumes on a single output volume, the Owner field for the output volume will be set from the Owner field of the input volume of the last data set copied to the output.

DUMP= Controls whether an ABEND dump will be produced for errors that don't usually produce a dump, such as U0100, U0888, U0913, and U0502 ABENDS.

YES – produce a dump for these ABENDs, if a SYSUDUMP or SYSABEND DD is provided

NO – do not produce a dump

ASK – ask the operator, via a WTOR, if a dump should be produced.

Delivered default(*): DUMP=NO

ECHO= Used to produce a display of the values of FATSCOPY options used by this job. It shows the values resulting from keywords you have specified, and other values used by default from the FATSCOPY Global Options Table.

NO – do not echo the FATSCOPY option values in SYSPRINT.

YES – echo the FATSCOPY option values in SYSPRINT.

Delivered default(*): ECHO=NO

EXCRPGM=(pgmname,…, pgmname) Used to exclude data sets from being copied if one of the *pgmnames* specified matches the name of the program that created the data set. For example, specifying EXCRPGM=(FDRDSF) will exclude any data set that was created by FDRDSF. The keyword requires a fully qualified program name, accepts up to 10 values, and can be used for either CATDSN or ALLDSN processing.

Delivered default: None

EXLASTJOB=(jobname,…, jobname) Used to exclude data sets from being copied if one of the *jobnames* specified matches the name of the job that last used the data set. Each jobname can be either a 1-8 alphanumeric character job name, or a 1-7 alphanumeric character job name prefix followed by an asterisk. For example, specifying EXLASTJOB=FATS* will exclude all data sets that were last used by jobs whose names start with FATS. This keyword accepts up to 10 values, and can be used for either CATDSN or ALLDSN processing.

This keyword is supported only when the tape management system is RMM or TLMS.

Delivered default: None

EXLASTPGM=(pgmname,…, pgmname) With RMM, used to exclude a data set from being copied if one of the *pgmnames* specified matches the name of the program that last used the data set.

With CA 1, used to exclude a data set from being copied if one of the *pgmnames* specified matches the name of the program that last used the volume (or multivolume set) containing the data set.

For example, EXLASTPGM=FATSCOPY will exclude all data sets that were last used by FATSCOPY. The keyword requires a fully qualified program name, accepts up to 10 values, and can be used for either CATDSN or ALLDSN processing. This can be used to avoid copying data sets that were already copied by FATSCOPY.

This keyword is supported only when the tape management system is RMM or CA 1.

Delivered default: None

EXPDTGROUP= "nnn" specifies the allowable spread of expiration dates when FATSCOPY is stacking data sets. "nnn" may be 0-999. The purpose of this keyword is to group together data sets that will expire within a few days of one another, so that one data set with an especially long retention will not cause the tape to be retained far past the expiration of other data sets on the tape. Data sets with special expirations (such as 99000 (catalog control) and 99nnn (cycle control)) will be grouped separately from data sets with explicit expirations (unless EXPDTGROUP=999 is specified).

EXPDTGROUP=999 is a special value that indicates that expiration dates will be ignored when stacking data sets, so that data sets with any expirations will be stacked together. When EXPDTGROUP=999 is specified, all data sets on the selected input tapes will be copied to a single output tape (or multi-volume tape set), except that the MAXFILE= operand may also be used to limit the number of input files to be copied to a given output tape. The value “999” is always honored regardless of the value of the GROUPBY= parameter.

When GROUPBY=VOLSET has also been used, other values of “nnn” are ignored; expiration dates are not used to group data sets. Each input volume (or multi-volume tape set) will be copied to a separate output tape or tape set.

If LASTAPE= is specified, EXPDTGROUP=999 is forced.

Delivered default(*): EXPDTGROUP=7

EXPIRED= **YES** – both expired and non-expired data sets (according to tape management data) may be selected.

NO – only data sets that are not expired in your tape management system may be selected.

Delivered default: EXPIRED=NO when SELECT CATDSN is used; EXPIRED=YES when SELECT ALLDSN is used.

GROUPBY= Specifies how the output data sets should be grouped. Each group of data sets will be used to create a separate output volume (or multivolume set).

VOLSET – Output data sets will always be grouped the same way they are grouped on the input volumes. All of the files selected from a single input volume (or multivolume set) will be grouped together on the output volume(s). GROUPBY=VOLSET minimizes the number of input tape mounts required. When GROUPBY=VOLSET is used, the value of EXPDTGROUP=nnn is ignored, except that EXPDTGROUP=999 is always honored.

EXPDATE – Output data sets will always be grouped by the expiration dates of the input files. The allowable spread of expiration dates used to group files together is the value specified by the EXPDTGROUP= parameter. GROUPBY=EXPDATE may result in many tape mounts for an individual input tape.

Delivered default: GROUPBY=VOLSET for SELECT ALLDSN; GROUPBY=EXPDATE for SELECT CATDSN.

HSMBAKMASK=
HSMMIGMASK=

If your installation does not use the default IBM data set names for your DFSMShsm (HSM) backup and migration data sets, you can specify masks or fully-qualified data set names that FATSCOPY can use to identify these data sets. See Section 32.5 for details on specifying a data set name mask.

By default, FATSCOPY will automatically exclude these backup and migration tape data sets. (If HSMML2=YES is specified, HSM ML2 data sets will not be excluded.)

Delivered default(*): the following masks will be used by FATSCOPY to identify and exclude HSM backup and migration data sets.

```
*.COPY.HMIGTAPE.DATASET
*.HMIGTAPE.DATASET
*.COPY.BACKTAPE.DATASET
*.DMP.*.V*.D+++++.T++++++
*.BACKTAPE.DATASET
*.DMP.T++++++.*.D+++++.V*
```

HSMML2=

Controls whether or not HSM Migration Level 2 data sets will be copied:

NO - HSM ML2 data sets will be excluded from copying.

YES - HSM ML2 data sets will be copied; an HSM RECYCLE command will be issued to copy any HSM ML2 data sets and update the HSM MCDS.

The rules for determining which data sets are considered to be HSM ML2 data sets are determined by the HSMMIGMASK parameter. If HSMMIGMASK has been specified, then the files considered to be HSM ML2 data sets are those that match the user-specified mask. If HSMMIGMASK is not specified, they are the ones that match the IBM default data set masks for migrated data sets, as listed above under the HSMMIGMASK= parameter.

Delivered default(*): HSMML2=NO

IMAGE

Used to create an image copy of a volume. An image copy is an exact bit-for-bit copy of an input tape, including the volume label, header and trailer labels, and all data files on the tape. The tape management information for the volume is updated so the information relates to the device where the volume is located after the image copy.

An image copy can be useful when you are migrating your tapes from one tape device to another, and want to maintain the original VOLSER and the position of each file on the volume. The original tape management information, such as, creation date, expiration date, etc., will be maintained; however, the device type, library name, storage group, etc. will be updated to reflect the new tape.

When using IMAGE, **do not specify EXPDT=98000 on the TAPEOUT DD statement**. The output volume **must** be under tape management control.

Many software products maintain databases that record the VOLSER, file sequence number, and/or block position of each file on the tape that they control. As long as the product using your tape does not record and use the device type for the tapes in its database (or you copy to the same device type as the input tapes), the image copies produced by FATSCOPY will be usable by the product on the output device.

You can use IMAGE only when the output device is an EMC DLm virtual tape system or a cartridge tape. **For more information about important considerations when using IMAGE, please see "Image Copies" in the Special Considerations section on page 32-31.**

IMAGE is valid only with SELECT ALLDSN.

LASTFILE= If files are being stacked on an output tape (MAXFILE= is greater than 1), LASTFILE= controls how FATSCOPY operates when the end of the output tape is reached. "nnnn" specifies a number of megabytes (MB). FATSCOPY will continue to add additional files to the current output tape until either MAXFILE= files have been written to the tape or the end of tape (EOT) is reached. If EOT occurs, FATSCOPY will calculate how many bytes from the current input file (the last file on the output tape) have been written. If the calculated megabytes is less than LASTFILE=, FATSCOPY will delete that file from the output tape, call for a fresh scratch output tape, and recopy the current input file to the output tape. If more than LASTFILE= megabytes have been copied from the current input file to the output tape, FATSCOPY will call for a fresh scratch tape and continue to copy that input tape, creating a multi-volume output tape.

LASTFILE= does not apply to the first output file on a given output tape volume; if the current input file completely fills the output tape, it will always create a multi-volume output file.

LASTFILE= is ignored if a SELECT ALLDSN statement is used to select data sets, unless EXPDTGROUP=999 is also specified.

Delivered default(*): LASTFILE=1000 which is 1000 Megabytes or 1 Gigabyte (GB).

LASTAPE= If files are being stacked on an output tape (MAXFILE= is greater than 1), LASTAPE= allows an output tape to be used in multiple FATSCOPY runs, adding additional files to the tape in each run. LASTAPE= specifies a valid fully-qualified data set name (up to 44 characters). The data set name may also be a GDG; specify the GDG base name followed by “(0)”. At the end of a FATSCOPY step, an empty file with this data set name will be created and cataloged, to mark the last file created on the tape (for a GDG, the +1 generation will be created). In a subsequent FATSCOPY step (even in a different job), FATSCOPY will determine if the data set is cataloged; if so, the previously-used tape will be mounted and repositioned to the LASTAPE file number so that additional files can be added to the tape after that point. If the LASTAPE data set is not cataloged, FATSCOPY will call for a fresh scratch tape. If you don’t want FATSCOPY to use a previous tape, simply uncatalog the LASTAPE name. LASTAPE= is mutually exclusive with the OUTFSEQ= operand.

When this parameter is specified, FATSCOPY will not allow a multi-volume output tape to be created, unless LASTMULT is also specified.

If you are using a DATACLAS or STORCLAS on your TAPEOUT DD statement to assign the output device, you must add UNIT=(,,DEFER) to the TAPEOUT DD statement to allow FATSCOPY to call for a mount of the correct output tape.

If LASTAPE= is specified, EXPDTGROUP=999 is forced.

LASTMULT If LASTAPE is specified, by default FATSCOPY will not allow a multi-volume output tape to be created. However, if LASTMULT is also specified, FATSCOPY will use the LASTFILE= value (or its default) to determine whether to create a multi-volume tape. LASTMULT is mutually exclusive with the THRESHOLD= operand.

LINECNT= "nnnnn" (1-32767) specifies the number of lines per page to print on all FATSCOPY print data sets.

Delivered default(*): LINECNT=60

LOGERROR= When using the system logger to write audit records, this specifies what FATSCOPY should do if it encounters an error when attempting to write an audit record to the log stream.

ABEND – Terminate the job with an abend.

CONTINUE – Write a warning message, but continue copying data sets.

Delivered default(*): LOGERROR=ABEND

MAJERR= "nnnnn" (1-32767) specifies the maximum number of major errors (non-data check errors) that are allowed to occur on a file being copied before FATSCOPY will stop copying the file. Non-data check tape errors are usually equipment check errors or "intervention required" (tape no longer ready) conditions; these may indicate problems with the input tape drive or with the current input tape. Exceeding “nnnnn” major errors on an input tape file will cause FATSCOPY to stop copying the current file. FATSCOPY will then reset the major error count to '0' and attempt to copy another selected tape file, if any remain.

Delivered default(*): MAJERR=1, stop copying the current file after the first major error.

MAXERR= "nnnnn" (1-32767) specifies the maximum number of data checks that are allowed to occur on a file being copied before FATSCOPY will stop attempting to copy the file. Data checks usually indicate an error on the current input tape, but can also indicate errors with the tape drive. Encountering “nnnnn” data checks on an input tape file will cause FATSCOPY to stop copying that file. FATSCOPY will then reset the data check count to '0' and attempt to copy another selected tape file, if any remain.

Delivered default(*): MAXERR=1, stop copying after the first data check.

MAXFILE= "nnnnn" (0-65535) specifies the maximum file number to create on the output tape. If MAXFILE= is not equal to 1, then tape stacking (writing multiple files on an output tape) will be done. Once “nnnnn” files have been written on an output tape it will be dismounted and a fresh scratch tape will be mounted to receive new copied files.

If MAXFILE=1, then no stacking will occur and a fresh scratch tape will be mounted for each output file.

If MAXFILE=0, tape stacking will be done. Files will be added to the output tape until it reaches its capacity, at which point a fresh scratch tape will be mounted to continue copying files.

Delivered default(*): MAXFILE=0

MAXRBLK= "nnnnnn" is a decimal number (4-262144) giving the largest input blocksize to be considered for reblocking. MAXRBLK= is meaningful only if REBLOCK= is also specified. Any input file whose blocksize on TAPEIN is greater than "n" will not be reblocked. .

Delivered default: All files are reblocked if REBLOCK= is specified.

MAXTERR= If "nnnnnn" temporary data checks are detected on TAPEIN, FATSCOPY will stop copying the current file, issue a non-zero return code, and start a new file. "n" may be 1-32767.

Delivered default(*): MAXTERR=100

MAXTOTFILE= (CATDSN only) "nnnnn" (1-65535) specifies the maximum number of input tape files to copy in this FATSCOPY step. Once “nnnnn” files have been selected for copying, FATSCOPY will stop searching the catalogs for additional files.

Delivered default(*): MAXTOTFILE=1000

MULTIFILE= (CATDSN only) Controls whether multi-file input tapes will be copied. When you are using FATSCOPY for tape stacking, this can be used to avoid copying tapes that already have stacked files (more than one file on the tape).

NO - tapes containing multiple files will not be copied.

YES - tapes containing multiple files will be copied. If you are using FATSCOPY to migrate tape files to a new type of tape, you will probably want to specify MULTIFILE=YES.

Delivered default(*): MULTIFILE=NO - only single-file tapes will be copied.

MULTIVOL= Controls whether multivolume files or sets will be copied.

YES – files that are located on single volume and multivolume sets will be copied.

NO – with SELECT ALLDSN, files that are located on multivolume sets will not be copied. With SELECT CATDSN, multivolume files will not be copied.

ONLY – with SELECT ALLDSN, only files that are located on multivolume sets will be copied. With SELECT CATDSN, only multivolume files will be copied.

For SELECT ALLDSN, MULTIVOL= controls file selection whether or not an individual file is a multivolume file. For example, a single-volume file which is located on a volume that is part of a multivolume set will be selected if MULTIVOL=YES or MULTIVOL=ONLY is specified, and bypassed if MULTIVOL=NO is used.

Delivered default(*): MULTIVOL=YES - both single volume and multi-volume tapes will be copied.

NOSUPULAB Specifies that user labels, if present, will be copied from the input file to the output file (copying of user labels will not be suppressed). NOSUPULAB cannot be used when SUPULAB has also been used.

Delivered default(*): NOSUPULAB

NOTCATRC= (ALLDSN only) This specifies the return code that will be issued by FATSCOPY when SELECT ALLDSN and CAT=ONLY are specified, and FATSCOPY is unable to catalog the output data set because the input wasn‘t cataloged. Allowed values are 0, 4, and 8.

Delivered default(*): NOTCATRC=0

OFFSITE= Controls whether tape volumes that are recorded in your tape management system as being off-site (vaulted) will be copied. Usually tapes that are off-site should not be copied because they are off-site and cannot be mounted. If you have a special reason for copying tapes that are marked as off-site and are sure they are available to be mounted, you can request that they be copied.

NO - off-site tapes will be automatically excluded.

YES - tapes marked as off-site will be selected for copying, as well as on-site tapes. Be sure that your SELECT statements will select only those off-site tapes that you know are actually available for mounting.

ONLY – only off-site tapes will be selected.

Note: Although FATSCOPY can copy offsite tapes, the output tapes will not have offsite status. You may need to use tape management utilities or manual updates to set the output tapes' status to offsite.

Delivered default(*): OFFSITE=NO

OPCAP= (used with SIMulate only) “nnnn” is a decimal number (1-9999) specifying the approximate uncompressed capacity of the output tapes, in gigabytes, that will be needed to copy the data sets selected during simulation. The output will display the approximate number of output tapes that will be required with no compression and with 3:1 compression, as well as a report showing the VOLSERs of the input tapes that will be needed for the copy job. For example, use OPCAP=10 if standard single-density 3590 tapes will be used for output.

Delivered default(*): OPCAP=10

OUTFSEQ= Specifies the file sequence number (1-65535) to be assigned to the first file written to TAPEOUT. This should be used only when adding files to an existing output tape; specify a value one larger than the last file on the tape. You must also specify the volser of the tape you want to append to on the TAPEOUT DD statement.

Note that values up to 9999 can also be specified by the LABEL= operand on the TAPEOUT DD statement. OUTFSEQ= is mutually exclusive with the LASTAPE= operand.

Delivered default: OUTFSEQ=1

REBLOCK= "nnnnnn" is a decimal number (4-262144) specifying the target blocksize for data reblocking. When this keyword is specified, input files with blocksizes less than “n” will be reblocked as they are copied to blocksizes as close to "n" as possible.

FATSCOPY only reblocks to larger blocksizes. Input files with blocksizes larger than “n” will not be reblocked. (FATAR may be used to reblock to smaller blocksizes.) MAXRBLK= may be specified to bypass reblocking on files that already have an acceptably large blocksize.

Delivered default: Blocks are copied exactly as read from the input tapes, without reblocking.

RESTART Used to restart a job using the data set created by a SIM job with the CHECKPT or a previous job that was stopped before all of the data sets selected for copying were copied. When RESTART is specified, FATSCOPY copies all of the data sets that weren't copied in the previous run. A DSNTABLE DD is required when RESTART is specified. All SELECT and EXCLUDE statements are ignored.

RETCODE RETCODE causes FATSCOPY to terminate with a return code (condition code) of 12 rather than a U0888 or U0300 abend when serious errors are found. RETCODE cannot be used when ABEND is also specified.

Note: When ECHO=YES has been used and RETCODE has been specified or used by default, this value will be shown in the ECHO output as

```
ABEND     ABNORMAL TERMINATION OPTION----------RETCODE
```

RETRY= (RMM only) Specifies the maximum number of times FATSCOPY will retry a failed RMM update when RMMFAIL=RETRY has also been specified. FATSCOPY will wait 30 seconds for each retry attempt. “nnn” may be 0-255.

Delivered default(*): RETRY=0

RMMFAIL= (RMM only) Determines how FATSCOPY is to proceed when an update of tape management information fails because RMM housekeeping happens to be running at the same time.

CONTINUE – FATSCOPY will issue a non-zero return code and an error message indicating that the RMM update failed, but continue copying any remaining data sets.

RETRY -- retry the failed update 'nnn' times based on the value of the RETRY= parameter. When the maximum number of retries is reached, FATSCOPY will issue a non-zero return code and an error message, and stop copying data sets. Note that, for each retry, FATSCOPY will wait 30 seconds.

STOP – immediately stop copying data sets after issuing a non-zero return code and an error message. (This works the same as RMMFAIL=RETRY with RETRY=0.)

Delivered default(*): RMMFAIL=CONTINUE

SELTERR= If FATSCOPY finds that a SELECT or EXCLUDE statement was never referenced (no data set on any input tape was selected by the statement):

YES -- a U0300 ABEND error will be issued at step termination to call attention to a possible control statement error.

NO – an ABEND error is **not** to be issued at step termination. A return code of 0 will be issued. You might use SELTERR=NO when you expect some unmatched SELECT/EXCLUDE statements, perhaps because some data sets may not exist.

Delivered default(*): SELTERR=YES

SHOWNEWN SHOWNEWN causes FATSCOPY to display in the Copy Report the new names of data sets assigned by the RENAME statement.

SORT= The SORT keyword is used only when SELECT CATDSN has been used to select data sets.

YES will cause the data sets to be sorted by expiration date group, VOLSER, and file sequence number after they have been selected, and to be copied in that sorted sequence. SORT=YES minimizes the number of input and output tape mounts required.

NO turns off sorting by expiration date group, VOLSER, and file sequence number. When SORT=NO is specified, data sets are copied in the sequence they appear in the SELECT statements. This is used when you want the data sets to be copied in a specific order. It may result in **many more input tape mounts** being required during the copy process. If EXPDTGROUP=999 is not specified, **many more output tape volumes** may also be required than if SORT=YES is used.

When SELECT ALLDSN is specified, the SORT keyword is ignored. Volumes are copied in the order in which they are listed in the SELECT statements (unless part of a multivolume set, in which case the volumes in the multivolume set are processed in the order they occur in the set).

Innovation strongly recommends using the default value of SORT=YES.

Delivered default(*): SORT=YES

SUPULAB Specifies that copying of user labels from the input file to the output file should be suppressed. If an input file has user labels, they will not be copied to the output file. SUPULAB cannot be used when NOSUPULAB has also been used.

This may be needed when you have an encryption product (such as CA Tape Encryption) that writes its own user labels that contain the file's encryption keys.

THRESHOLD= If LASTAPE= is specified, then THRESHOLD= specifies a percentage (1 to 99) used to determine when FATSCOPY will add another file to the current output tape. Before opening each new output file, the THRESHOLD= value is compared to the percentage used of the current volume, as reported by your tape management system. If the threshold is exceeded, FATSCOPY will close the current tape and call for a fresh scratch tape. Note that the percentage used may not be recorded by your tape management system for all types of tape. THRESHOLD= is mutually exclusive with the LASTMULT operand.

The THRESHOLD keyword is not supported when the tape management system is TLMS.

TMSDATA= Specifies whether tape management data from each input file is to be copied to the output file after the input file is successfully copied.

COPY will copy the tape management data from the input file to the output file. This data includes: creation date, creating job name, and creating step name.

NOCOPY not copy the tape management data. The tape management records for the output file will show the current date as the creation date, and the FATSCOPY job/step as the creating job/step.

Delivered default(*): TMSDATA=COPY

TMSIN= **This keyword should be used ONLY if you have more than 1 active tape management system and you are copying from tapes controlled by CA 1 to tapes controlled by RMM or vice versa.**

FATSCOPY will normally determine the tape management system used for the input tapes by testing which FATS tape management interface you have installed, but if you installed both the DFSMSrmm and CA 1 interfaces, you may specify which system controls the tapes you are copying:

RMM - the input tape management system is IBM's DFSMSrmm.

CA1 - the input tape management system is CA Technologies' CA 1 (TMS).

Delivered default: If both interfaces are installed, DFSMSrmm is assumed to be the input tape management system. Otherwise, the installed interface is used.

TMSINPUT= Specifies how the input tape files are to be handled by your tape management system after they are successfully copied.

KEEP will retain the input files. *Note:* if the retention of the file is "catalog control" and the file is recataloged to the output tape (via CAT=RECAT), the tape may still be scratched.

SCRATCH will expire each input file after it is copied. If all files on an input tape are scratched, your tape management system will return the tape to scratch status. By default, FATSCOPY will retain a volume 4 days before it is scratched. You can set this value higher or lower using the ADDXDAYS keyword.

When TLMS or CA 1 is the tape management system, special processing is used when scratching the control data set (TLMS) or first data set (CA 1). Since scratching this data set results in the **entire volume** (or multivolume set) being expired **regardless** of the expiration dates of the other data sets on the volume, FATSCOPY sets the expiration date of this data set to the highest expiration date of all the unexpired data sets remaining on the volume/ multivolume set. If FATSCOPY successfully copies all the files on the volume/multivolume set, it then goes back and resets the expiration date of the control data set (TLMS) or first data set (CA 1) to the scratch date.

Delivered default(*): TMSINPUT=KEEP

TMSOUT= **This keyword should be used ONLY if you have more than 1 active tape management system, and you are copying from tapes controlled by CA 1 to tapes controlled by RMM or vice versa.**

Specifies that the tape management system used to manage the output tapes is **different** from the one used to manage the input tapes (from which FATSCOPY acquired tape information).

RMM - the output tape management system is IBM's DFSMSrmm.

CA1 - the output tape management system is CA Technologies' CA 1 (TMS).

Delivered default: The output tape management system is the same as the input tape management system.

UNITIN= Specifies an esoteric (e.g., UNITIN=TAPE), generic (e.g., UNITIN=3490) or specific (e.g., UNITIN=3A0) unit name to be used to dynamically allocate input tapes when SELECT ALLDSN,VOL=volser is specified. A leading slash is required when specifying a specific 4-digit tape device address (e.g., UNITIN=/14B0). The name can be any name that can be used in the JCL parameter UNIT=unitname. UNITIN= can be used whenever a special unit name is required to allocate the proper tape drive to the input tapes selected in this step, such as a drive in a tape library.

Delivered default: FATSCOPY will query the "last used unit" (last tape device on which the tape was mounted) from your tape management system, determine the generic device type of that unit (e.g., 3590-1) and use that generic unit for dynamic allocation.

VIRTTYPE= When using a Virtual Tape Server backed by physical tapes, specifies the type of VTS that contains the volumes to be copied. This keyword is used only when the PHYSVOL or ALLPHYS keyword is used on a SELECT statement, in which case VIRTTYPE= is required.

IBM - The tape server's BVIR function will be used to determine all the logical volumes contained on the specified physical volume(s). When VIRTTYPE=IBM is used, the MAPTAPE DD statement is also required.

STK - The STK/Oracle Host Software Component (HSC) API will be used to determine all the logical volumes contained on the specified physical volume(s).

Delivered default(*): none

VIRTUNIT= Specifies the esoteric name (1-8 characters) assigned to CopyCross virtual tape drives, for example, VIRTUNIT=VT3590. The DEVTYPE=COPYCROS parameter must also be specified on the SELECT control statement.

32.3 SELECT/EXCLUDE STATEMENTS

The FATSCOPY input must include one or more SELECT statements to select data sets from your system catalogs or tape volumes from your tape management data. (Exception: a COPY job with the RESTART keyword does not use a SELECT statement. The selections in that case are taken from the DSNTABLE data set.)

Each SELECT statement **must** contain either a CATDSN= or an ALLDSN operand**. You cannot use both CATDSN= and ALLDSN on SELECT statements in the same FATSCOPY job**.

SELECT STATEMENTS with CATDSN

SELECT CATDSN=dsname|mask

Additional optional operand:

,VOL=(volser1,…,volser20)

SELECT STATEMENTS with ALLDSN

SELECT ALLDSN

Exactly one of the following additional operands is required with each ALLDSN:

,VOL=volser

,PHYSVOL=physicalvolser

,ALLPHYS=logicalvolser

You are allowed to have SELECT ALLDSN statements with different additional operands in a single job; for example,

```
SELECT ALLDSN,VOL=001435
SELECT ALLDSN,PHYSVOL=P00001
```

ADDITIONAL OPERANDS FOR SELECT

The following optional operands may be added to SELECT statements with either CATDSN= or ALLDSN.

,CRDATE=yyyyddd | yyyy.ddd

,CRDAYS=nnnn

,CRJOB=(jobname1,…,jobname20)

,DSSIZE=nnnnn

,EXPDATE=yyyyddd | yyyy.ddd

,FILESEQ=nnnnn

,LOCATION=(location1,…,location5)

,OUTCODE=(code1,…,code5)

The following optional operand may be added to SELECT statements when VOL= is used with ALLDSN.

,**NUMVOLS=**nnnn

The following optional operand may be added to SELECT statements when VOL= is used with CATDSN.

,CATLIMITGDG=nnn|-nnn

,DEVTYPE=(devtype1,…,devtype5)

A data set must meet **all** of the selection operands specified on an individual SELECT statement in order for it to be selected. If there are multiple SELECT statements, a data set needs to match the criteria for only one of the SELECT statements in order to be selected.

When CATDSN= is specified, you provide the names (using either fully-qualified names or name filters) of the data sets to be selected from the **z/OS system catalogs** and may also specify other operands to limit the data set selections. Only data sets cataloged to a tape device type will be selected. The CATDSN= operand, and the VOL=, DEVTYPE=, and/or FILESEQ= operands, if they are specified, will be used to select data sets from your system catalogs. Additional operands, if specified, will be used to further filter the data sets selected from the catalog using information about those data sets from your tape management system. Note that if you specify the tape management-related operands, the data sets will be selected only if the information for the data set in the catalog matches the corresponding information for the data set in tape management records.

Note that FATSCOPY only copies data sets in their entirety. If VOL= is specified and all pieces of a multivolume data set do not match the volumes in the parameter list, the data set will not be selected.

When ALLDSN is specified, it must be accompanied by a **single** VOL=, PHYSVOL=, or ALLPHYS= operand. It may also be optionally accompanied by a combination of FILESEQ=, CRDATE=, CRDAYS=, EXPDATE=, CRJOB=, and DSSIZE= operands. The optional NUMVOLS= operand may also be used with VOL=.

- **VOL=** specifies a single tape volume serial that will be queried against your **tape management database**. All data sets on that input volume will be selected for copying (unless restricted by additional operands, as described below). If the tape volume is part of a multi-volume tape set, all data sets on <u>all volumes in that set</u> will be selected for copying.

  If you add the optional NUMVOLS=nnnn parameter, a range of nnnn volumes beginning with the value specified by VOL= will be queried, and data sets on those volumes will be selected. If any volume in that range is part of a multivolume set, all volumes of the set are selected, including those which are not in the specified range.

- **PHYSVOL=** specifies a single *physical* tape volume on a tape-backed virtual tape system (VTS) that will be queried against the **VTS's internal database** to get a list of logical volumes on that physical tape; those *logical* volumes will be queried against your tape management database. All data sets on the logical volumes on that physical tape will be selected for copying (unless restricted by additional operands, as described below).

- **ALLPHYS=** is similar to PHYSVOL=, except that you specify a *logical* tape volume in a tape-backed VTS. The *physical* tape containing that logical volume is determined from the VTS's internal database; then all the logical volumes on that physical tape are selected for copying (unless restricted by additional operands, as described below).

If any additional optional operands are specified on the SELECT ALLDSN statement, they will be used to further filter the data sets on the selected volume using information about those data sets from your tape management system.

EXCLUDE STATEMENT

EXCLUDE CATDSN=dsname|mask or **DSN=**dsname|mask
E

,VOL=(volser1,…,volser20)

,DEVTYPE=(devtype1,…,devtype5)

You may optionally include one or more EXCLUDE statements to exclude certain data sets that would otherwise be selected by a SELECT statement that follows it. **For ALLDSN, all the EXCLUDE statements must precede the *first* SELECT statement. For CATDSN, the EXCLUDE statements must precede the SELECT statements *to which they apply***. EXCLUDE statements apply to *all* following SELECT statements.

For example,

```
EXCLUDE CATDSN=JAT.PROD.FILE
SELECT CATDSN=JAT.PROD.**,VOL=ABC009
SELECT CATDSN=JAT.PROD.**,VOL=XYX009
```

will exclude any data sets named JAT.PROD.FILE on either volume ABC009 or XYX009, while

```
SELECT CATDSN=JAT.PROD.FILE,VOL=ABC009
EXCLUDE CATDSN=JAT.PROD.FILE
SELECT CATDSN=JAT.PROD.FILE,VOL=XYX009
```

will allow JAT.PROD.FILE to be selected from volume ABC009, while it will not be selected from volume XYX009.

The EXCLUDE statement **must** contain **one** of the following:

- When the EXCLUDE is being applied to a SELECT CATDSN statement, the EXCLUDE **must** contain a CATDSN= operand. In this case, it may also optionally contain VOL= and DEVTYPE= operands to limit the exclusion.
- When the EXCLUDE is being applied to a SELECT ALLDSN statement, the EXCLUDE **must** contain a DSN= operand. EXCLUDE DSN= will accept a fully qualified DSN or a data set mask to exclude data sets that would normally be selected by a SELECT ALLDSN statement. **The DSN= operand is valid only for EXCLUDE**; it is not valid for SELECT.

SELECT/ EXCLUDE COMPARISON OPERATORS

Although the syntax listings above show that all operands are followed by an equal sign (=), certain operands may be followed by comparison operator from the list below. Each operator has two forms, one or two special characters (such as = or >=) or an alphabetic equivalent surrounded by periods (such as .EQ. or .GE.). The operand descriptions that follow indicate which operands support which comparison operators.

=	or	.EQ.	equal
¬=	or	.NE.	not equal
<	or	.LT.	less than
>	or	.GT.	greater than
<=	or	.LE.	less than or equal to
>=	or	.GE.	greater than or equal to

When an operand allows a range to be specified by using the operand twice, the range must be defined by a pair of “greater than” (or “greater than or equal to”) and “less than” (or “less than or equal to”) comparisons.

COMPARISON OPERATOR EXAMPLES

```
SELECT  CATDSN=ABC.**
```

will select all cataloged tape data sets starting with "ABC."

```
SELECT  CATDSN=XYZ.**.MASTER,VOL=B01*
```

will select all tape data sets starting with "XYZ." and ending in ".MASTER" that are cataloged to tape volumes starting with B01

```
EXCLUDE  CATDSN=PAYROLL.HISTORY.**
SELECT   CATDSN=PAYROLL.**
```

will select all cataloged tape data sets starting with "PAYROLL." except those starting with "PAYROLL.HISTORY."

```
SELECT   CATDSN=ABC.**,CRDATE.GE.2009001,CRDATE.LE.2010365
```

will select all cataloged data sets starting with ABC. whose creation date is in 2009 or 2010.

```
SELECT   CATDSN=XYZ.**,DSSIZE>25,DSSIZE<=75
```

will select all cataloged data sets starting with XYZ. whose size is from 26 to 75 megabytes.

```
EXCLUDE  DSN=SYS9.**
SELECT   ALLDSN,VOL=3011
```

will select all the data set on volume 003011 except those data sets with a high level qualifier of SYS9.

SELECT/ EXCLUDE OPERANDS

The following are the operands for the SELECT and EXCLUDE statements.

CATDSN=

CATDSN= specifies a fully-qualified data set name or a filter to be used for generic data set selection from the z/OS system catalogs, as described in detail in Section 32.5. CATDSN supports only the equal operator (= or .EQ.). When specified on a SELECT statement, it causes the z/OS system catalogs to be searched for data sets that match. When specified on an EXCLUDE statement, it specifies data sets that will be excluded from the catalog search on SELECT statements that follow it.

If a fully-qualified name is specified, that name will be located in the z/OS system catalogs. Specification of a relative generation number for GDG data sets is supported (e.g., CATDSN=A.B(-1)).

If a filter is specified on a SELECT statement, then catalogs will be scanned for all cataloged data sets matching the filter (unless they are excluded by a preceding EXCLUDE statement).

CATDSN=** will select all cataloged tape data sets. This should be used only when other operands are also specified to limit the data sets selected.

CATDSN=** or any filter starting with a non-specific selection character will cause **all** catalogs on your system to be scanned, which may take a considerable amount of time.

CATDSN= cannot be used when doing an image copy.

Additional considerations for the CATDSN=filter are explained in Section 32.5.

ALLDSN

ALLDSN must be followed by a VOL=, PHYSVOL=, or ALLPHYS= that specifies a single tape volume serial.

- For VOL=, the specified tape volume will be looked up in your tape management database; all data sets on that volume will be selected. If you add NUMVOLS=, a range of volumes will be looked up, and data sets on those volumes will be selected. If a volume is part of a multivolume set, data sets on all volumes in the set will be selected.
- For PHYSVOL=, the specified VTS *physical* tape volume will be looked up in the VTS's internal database; all the *logical* volumes on that physical tape will be determined, and all data sets on those logical volumes will be selected.
- For ALLPHYS=, the specified *logical* tape volume will be looked up in the VTS's internal database to determine which *physical* volume it is located on. Then, that physical volume is looked up in the VTS's database to determine *all* the logical volumes on that physical tape; all data sets on those logical volumes will be selected.

The ALLDSN operand is valid only for SELECT; it is not valid for EXCLUDE.

Note that SELECT ALLDSN will automatically bypass the first file on an input tape if it has an invalid data set name (such as the name created by CA 1 COPYCAT).

DSN=

Specifies a fully qualified data set name or a data set mask to exclude data sets that would normally be selected by a SELECT ALLDSN statement. **The DSN= operand is valid only for EXCLUDE**; it is not valid for SELECT.

VOL=
PHYSVOL=
ALLPHYS=

One of these 3 keywords are used on a SELECT statement to specify a tape volume serial or a volume serial prefix to be selected. Only one of them can be used on each SELECT statement.

- **VOL=** specifies a logical volume serial or volume serial prefix to be selected. This can be used with either SELECT CATDSN or SELECT ALLDSN. This volume serial will be looked up in your tape management database and all data sets on that tape will be selected for copying, unless further restricted by other SELECT or EXCLUDE operands. If it is part of a multi-volume tape set, all tapes in the set will be copied; you can specify any volume in the tape set. If you specify more than one volume of a multivolume set in a single job, the set will be selected only once. If you also specify NUMVOLS=, then a range of volumes will be looked up, and data sets on those volumes will be selected.
- **PHYSVOL=** specifies a *physical* tape volume serial on a tape-backed virtual tape system. It will be looked up in the VTS's internal database to determine all of the logical volumes on that physical tape. Those logical volsers will be looked up in your tape management database, and all data sets on those logical volumes will selected for copying, unless further restricted by other SELECT/EXCLUDE operands. This is intended as an aid in migrating entire physical volumes from a VTS, when you want to identify a tape to copy by its physical volser.
- **ALLPHYS=** specifies a *logical* tape volume serial on a tape-backed virtual tape system. FATSCOPY will use the VTS's internal database to determine which physical tape the logical volume is located on. Processing continues as if PHYSVOL= had been specified for that physical volume. This is intended as an aid in migrating entire physical volumes from a VTS, when you want to identify a tape to copy by the volser of one of the logical volumes on the tape.

VOL= can be used with either SELECT CATDSN or SELECT ALLDSN. PHYSVOL= and ALLPHYS= can be used only with SELECT ALLDSN.

When PHYSVOL= or ALLPHYS= is used, the VIRTTYPE= keyword on the COPY statement must also be used to identify the type of virtual tape system used. For VIRTTYPE=IBM, the MAPTAPE DD statement is also required. For VIRTTYPE=STK, the HSC link library JOBLIB DD statement is also required, if this library is not in the linklist.

***Note**:* for a StorageTek VTS, a logical volume will be included only if it is the **first** migrated copy of the volume. (Identical copies of a logical volume may be migrated to up to 4 physical volumes within the VTS. Second, third, and fourth migrated copies of a logical volume will not be selected by FATSCOPY.)

If ALLDSN is specified, then one of VOL=vvvvvv, PHYSVOL=vvvvvv, or ALLPHYS=vvvvvv is **required** and must specify a **single** tape volume serial.

If CATDSN= is specified, then VOL=vvvvvv is optional; only data sets cataloged to the volume(s) specified will be selected. For a multivolume data set, if any volume of the data set does not match the volume(s) specified, the data set will not be selected. The VOL operand supports all 5 comparison operators shown on page 32-19. You can specify:

`vvvvvv`	a single tape volume serial, 1-6 alphanumeric characters
`vvvvv*`	a volume serial prefix (1-5 alphanumeric characters) followed by an asterisk

You can also specify a range of volume serials from a low value to a high value by specifying the VOL operand twice on the same SELECT or EXCLUDE statement. For example,

```
VOL>=400000,VOL<=444999
```

A list of volume serials or prefixes can be specified, enclosed in parentheses and separated by commas; the data set will be selected if it matches any one of the values specified. Up to 20 volume serials can be specified. In this case, only the equal operand (= or .EQ.) is supported.

If VOL is omitted, then the volume serial will not be considered when processing the SELECT or EXCLUDE statement.

CATLIMITGDG= (With CATDSN only) "nnn" or "-nnn" is used to limit the number of generations that CATDSN= will select from a GDG (generation data group), as described in "Selecting Data Sets from the Catalog" on page 32-29. The maximum value of nnn is 255.

CRDATE= Specifies a creation date as a Julian date in the format yyyyddd or yyyy.ddd. Only data sets created on the date(s) specified will be selected. CRDATE will be honored only if the data set selected from the catalog is found in your tape management records. The CRDATE operand supports all 5 comparison operators shown on page 32-19.

You can also specify a range of creation dates from a low value to a high value by specifying the CRDATE operand twice on the same SELECT statement. For example,

```
CRDATE>=2011001,CRDATE<=2011159
```

You cannot specify both CRDATE and CRDAYS on the same SELECT/EXCLUDE statement.

CRDAYS= Only data sets created "nnnn" days ago, relative to the current date, will be selected for copying. CRDAYS will be honored only if the data set selected from the catalog is found in your tape management records. The CRDAYS operand supports all 5 comparison operators shown on page 32-19.

You can also specify a range of days from a low value to a high value by specifying the CRDAYS operand twice on the same SELECT statement. For example,

`CRDAYS=7` selects data sets created exactly 7 days ago

`CRDAYS>7` selects data sets created more than 7 days ago

`CRDAYS<7` selects data sets created less than 7 days ago (i.e. within the past 6 days)

`CRDAYS.GE.7,CRDAYS.LE.30` selects data sets created 7 to 30 days ago

`CRDAYS=0` selects data sets created today

You cannot specify both CRDAYS and CRDATE on the same SELECT/EXCLUDE statement.

CRJOB= Specifies a creating job name (1 to 8 characters) or job name prefix. Only data sets that were created by this job, as recorded in your tape management system, will be selected. CRJOB will be honored only if the FATS tape management interface is enabled, and the data set selected from the catalog is found in your tape management records. If multiple job names are specified, enclose them in parentheses, separated by commas; a data set will be selected if it matches any one of the values specified. Up to 20 job names can be specified. The CRJOB operand supports only the equal operator (= or .EQ.). You can specify:

`jjjjjjjj` a single job name, 1-8 alphanumeric characters

`jjjjjjj*` a job name prefix (1-7 alphanumeric characters) followed by an asterisk

DEVTYPE= (With CATDSN only) Specifies a tape device type. Only data sets cataloged to the device type specified will be selected. DEVTYPE= must specify a valid z/OS generic tape device type, such as 3480, 3490, or 3590-1. You cannot use esoteric names (such as TAPE or CART) since these names are not recorded in the system catalogs. If multiple device types are specified, enclose them in parentheses, separated by commas; a data set will be selected if it matches any one of the values specified. Up to 5 device types can be specified. DEVTYPE supports only the equal operator (= or .EQ.).

If DEVTYPE=COPYCROS is specified, then the value specified for the VIRTUNIT= operand on the COPY statement is used to specify the esoteric name to be used to select data sets cataloged to CopyCross devices.

DSSIZE= Specifies a data set size, in megabytes. Only data sets that have this size, estimated from the blocksize and block count recorded in your tape management system, will be selected. “nnnnn” must be less than 65536.

You can also specify a range of sizes from a low value to a high value by specifying the DSSIZE operand twice on the same SELECT statement. For example,

```
DSSIZE>=2,DSSIZE<20
```

EXPDATE= Specifies an expiration date as a Julian date in the format yyyyddd or yyyy.ddd. Only data sets whose current expiration dates match this value, as recorded in your tape management system, will be selected. EXPDATE will be honored only if the data set selected from the catalog is found in your tape management records. The EXPDATE operand supports all 5 comparison operators shown on page 32-19.

You can also specify a range of creation dates from a low value to a high value by specifying the EXPDATE operand twice on the same SELECT statement. For example,

```
EXPDATE>=2012001,EXPDATE<=2012.159
```

FILESEQ= Specifies a file sequence number. This is the file number of the file on a multi-file tape, originally specified by the LABEL=nnnn JCL parameter when the file was created. Only data sets with file number "nnnnn" (1-65535) will be selected. The FILESEQ operand supports all 5 comparison operators shown on page 32-19.

You can also specify a range of file sequence numbers from a low value to a high value by specifying the FILESEQ operand twice on the same SELECT statement. For example,

```
FILESEQ>1,FILESEQ<123
```

On a SELECT ALLDSN statement, you can use FILESEQ to select a specific file or a range of file numbers, etc. This can be used to restart FATSCOPY when it terminated prematurely for some reason, specifying FILESEQ> the last file number successfully copied. For example,

```
SELECT ALLDSN,VOL=123456,FILESEQ>122
```

LOCATION= (RMM and TLMS only) Data sets whose Current Location values match one of these specified values will be selected for copying. Up to 5 values may be specified by enclosing them in parentheses; if only one value is specified, the parentheses may be omitted. Each Location value can be up to 8 characters in length. LOCATION accepts only the equal operator (= or .EQ.).

NUMVOLS= (With ALLDSN,VOL= only) Specifies a number nnnn from 1 to 1000. This causes a range of nnnn consecutive volumes, beginning with the volume specified by the VOL= parameter, to be looked up in your tape management system. Data sets located on that range of volumes will be selected. If a volume in the range is part of a multivolume set, all volumes of the set will be selected – including volumes which are not within the specified range.

This is the equivalent of specifying ALLDSN statements for nnnn consecutive volumes. For example,

```
SELECT ALLDSN,VOL=A40006,NUMVOLS=5
```

gives exactly the same selection results as

```
SELECT ALLDSN,VOL=A40006
SELECT ALLDSN,VOL=A40007
SELECT ALLDSN,VOL=A40008
SELECT ALLDSN,VOL=A40009
SELECT ALLDSN,VOL=A40010
```

OUTCODE= (CA 1 only) Data sets whose CA 1 Outcode values match one of these specified values will be selected for copying. Up to 5 values may be specified by enclosing them in parentheses; if only one value is specified, the parentheses may be omitted. Each Outcode value can be up to 4 characters in length; if a value contains imbedded blanks, the value must be coded within single quotation marks (e.g. 'NJ D') . OUTCODE accepts only the equal operator (= or .EQ.).

32.4 RENAME STATEMENT

RENAME STATEMENT

RENAME LF=ALL,NEWI=newindexmask

RENAME can be used to change the name of the file(s) created on the output tape.

NEWI= specifies that the output data set name is to be constructed by adding or replacing one or more index levels in the original name from the input tape; replacement index levels do not need to be the same length as the original indexes they replace. In the simplest case, FATSCOPY will use each index level specified in NEWI in place of the corresponding index in the original name. Any remaining index levels at the end of the name are copied unchanged. This can be used to easily change the first indexes of the name.

For example, if copying data set A.B.C.D,

`NEWI=D`	results in	`D.B.C.D`	(first index replaced)
`NEWI=DD.E`	results in	`DD.E.C.D`	(first 2 indexes replaced)

If a period is specified without any preceding characters, FATSCOPY will copy one original index level from the input data set name to the output. This allows you to easily modify indexes in the middle of the name.

For example, if copying data set A.B.C.D,

`NEWI=..E`	results in	`A.B.E.D`	(third index replaced)
`NEWI=FF...G`	results in	`FF.B.C.G`	(first and fourth indexes replaced)

If + is specified before a new index level, FATSCOPY will insert that new index into the output data set name at that point. If ++ precedes the new index, it will be added to the end of the name. If − is specified, the next input index level will be dropped from (not copied to) the output name.

For example, if copying data set A.B.C.D,

`NEWI=+F`	results in	`F.A.B.C.D`	(new first index added)
`NEWI=..+F`	results in	`A.B.F.C.D`	(new third index added)
`NEWI=++F`	results in	`A.B.C.D.F`	(new last index added)
`NEWI=..-`	results in	`A.B.D`	(third index dropped)
`NEWI=Q.-.+E`	results in	`Q.C.E.D`	(combination)

Note that, except for the ++ option, every period in the NEWI= mask corresponds to a period (one index level) in the original (input) data set name. FATSCOPY will check the resulting new name to insure it meets IBM standards.

If the NEWI= value ends in a GDG relative generation number, e.g., NEWI=..NEWMAST(-2), that relative number will be added to the end of the newname, and a LOCATE done to get the proper absolute generation number.

32.5 DATA SET NAME FILTERS (MASKS)

The CATDSN= operand allows to you specify a filter (sometimes referred to as a “mask”) that will be applied to data sets selected from system catalogs. The filter allows you a great deal of flexibility in specifying the names to be selected.

FILTER CHARACTERS

The data set name filter is a string of characters that specify the data sets to be selected. Each character in the filter may represent a single character in the name of the data sets or may represent a variable number of characters. The filter characters may be:

- any alphanumeric character (A-Z or 0-9) or national character ($ # @ in the U.S.) represents itself and must appear in the indicated position in the data set name
- / (slash) or % (percent) represents ANY single valid alphanumeric or national character
- | (vertical bar) represents any single alphabetic (A-Z) character
- + (plus) represents any single numeric (0-9) character
- ? (question) represents any single national ($ # @ in the U.S.) character
- . (period) is used to separate index levels.
- * (asterisk) by itself as an index level indicates that the index level must exist in the selected name, but that it can contain any valid characters and can be any valid length (1 to 8 characters). However, if a single asterisk is combined with other characters in an index level, then it represents a variable length string (zero or more characters) at that point in the index level
- `**` (double asterisk with no trailing period) represents zero or more characters or index levels. The filter routine will try to apply the remainder of the filter to the data set name beginning with the current character and stepping through the name until it matches or until the end of the name is reached
- `**.` (double asterisk with a trailing period) also represents zero or more characters or index levels, but the filter routine will try to apply the remainder of the filter to the data set name at the beginning of each index level. If not currently at the beginning of a level, it will start at the next level.

These special rules apply to ** with periods:

- `**.` at the beginning of the filter represents zero or more whole index levels at the beginning of the name
- `.**` at the end of the filter represents zero or more whole index levels at the end of the name.

The examples below are designed to illustrate the power and flexibility you have in using generic data set name selection.

EXAMPLE 1 `ACCOUNTS.PAY**`

will select data set beginning with "ACCOUNTS.PAY" such as

ACCOUNTS.PAY.OVERDUE
ACCOUNTS.PAYROLL.CHECKS

EXAMPLE 2 `USER1.*.CNTL`

will select any 3-level data set name with USER1 as the first level and CNTL as the third, such as

USER1.JCL.CNTL
USER1.TEST.CNTL

EXAMPLE 3 `USER+.**.*LIST`

will select any data set whose first index is USERn (n is numeric) and whose last index level ends in LIST (including LIST by itself) with any number of index levels between them, such as

USER1.LIST
USER3.ISPF.OUTLIST
USER9.TEST.PRINT.MYLIST

EXAMPLE 4 `**TEST**`

will select any data set with the string TEST anywhere in the name, such as

ABC.TEST.GHI.XYZ
TEST2.LIST
REPORT.CURRENT.TEST
USER3.MYTEST

EXAMPLE 5 `**A**?**B**`

will select any data set whose name contains an A, a national character, and a B, in that order, such as

SYS1.A.X#B
APPLIC1.A@B.LOAD
APPLIC2.MASTER.BALANCE$.BOUNCE

EXAMPLE 6 `PAYROLL.%%||+*.FILE`

will select any three-level data set whose first index is PAYROLL, whose second index consists of any 2 characters, followed by any 2 alphabetic characters, followed by any 1 numeric character, and up to 3 more characters, and whose third index is FILE, such as

PAYROLL.DCLX5.FILE
PAYROLL.A4TV3LM.FILE

EXAMPLE 7 `PAYROLL.//**.FILE.**`

will select any data set whose first index is PAYROLL, whose second index consists of at least 2 characters, and has any number of following index levels, one of which must be FILE, such as

PAYROLL.DCLX5.FILE
PAYROLL.AB.FACTORY1.FILE.OCT90
PAYROLL.FACTORY2.HISTORY.FILE

EXAMPLE 8 `**.G++++V00`

will select all generations of any GDG (Generation Data Group).

EXAMPLE 9 `**MASTER(-1)`

will select the -1 generation of any GDG whose name ends in MASTER, such as

PROD1.WEEKLY.PROD.MASTER(-1)
FINANCE.GLEDGER.XMASTER(-1)

This will only work when selecting data sets from the catalog, such as with the CATDSN= operand.

EXAMPLE 10 `**`

will select all data sets. *Note*: this may require a long time to conduct an extensive catalog search.

SELECTING DATA SETS FROM THE CATALOG

When generic data set selection is used to select data sets from system catalogs, one or more system catalogs will be searched depending on the filter specified.

By default, the search will start with the master catalog; if aliases are encountered that match the filter, the associated user catalogs will be searched as well. If there are enough explicit characters at the beginning of the filter, the search may be limited to one or a small number of user catalogs. For example, a filter of AB** means that only those user catalogs associated with aliases beginning with AB need be searched. However, if the filter begins with generic characters (such as **AB), then every user catalog with an associated alias in the master catalog must be searched. Although the catalog search is designed to be as efficient as possible, reading the catalogs directly whenever possible, this may still be a time-consuming process.

GDGs (generation data groups) are treated specially when selected from a catalog:

1) The filter will be compared to the GDG base name (without the GnnnnVnn absolute generation). If it matches the base, then all generations in that GDG will be selected, unless limited by CATLIMITGDG= as shown below.
2) The filter will also be compared to the full name of each generation (including the GnnnnVnn). All generations that match the filter will be selected.
3) If the filter ends in a relative generation number in parentheses (e.g., "PAYROLL.**(-n)" or "PAYROLL.**(0)") then only that generation will be selected from any GDGs selected; the filter will not select any non-GDGs.
4) You can specify CATLIMITGDG=-n and only the (-n)th generation of any GDGs that match the filter will be selected. However, the filter may also select non-GDG data sets.
5) You can also specify CATLIMITGDG=n and only the most recently created "n" generations of any GDGs that match the filter will be selected, The filter may also select non-GDG data sets.
6) For SMS-managed GDGs, any generations in rolled-off or deferred roll-in status will NOT be considered part of the GDG for selection; they will be treated as normal non-GDG data sets.

It is possible to assign aliases to non-VSAM data sets. These aliases are complete alternate names by which a data set can be referenced and are not related to the aliases assigned to catalogs to control which data sets will be in a given catalog. The filter will never select alias names; it is tested only against the true name of every data set.

If an alias is encountered in a master catalog that points to a user catalog on a volume that is not currently mounted (or does not exist) that catalog will be bypassed.

32.6 FATSCOPY SPECIAL CONSIDERATIONS

CHECKPOINT/ RESTART

FATSCOPY includes an option to create a checkpoint data set during a simulation run that can be used as input to a subsequent FATSCOPY job to copy the data sets selected during the simulation. FATSCOPY also has an Operator Communications function that allows you to display the STATUS of a FATSCOPY copy job or to stop the copy job (with an operator STOP or CANCEL command) and restart it at a later time. Both the CHECKPT and RESTART keywords and the Operator Communication require a DSNTABLE DD in the JCL.

If the DSNTABLE DD is present in the JCL, and both SIM and CHECKPT are specified (or the job is interrupted with a STOP or CANCEL command), FATSCOPY will create a data set containing an entry for each data set selected for copying. This data set is used as input to a subsequent FATSCOPY job that will copy all of the data sets selected during the simulation. When RESTART is specified, FATSCOPY uses the DSNTABLE data set to copy any data sets that weren't copied in the previous job using that DSNTABLE data set. When RESTART is specified, any SELECT statements will be ignored.

OPERATOR COMMANDS

You can display the total number of files selected for copying and the current file being copied by issuing a MODIFY (F) command through an operator console. The format of the command is:

F jobname,STATUS

You can also stop a FATSCOPY copy job when it finishes copying the current file by issuing a MODIFY (F) command through an operator console. The format of the command is:

F jobname,STOP

If a DSNTABLE DD statement has been used, FATSCOPY will create a data set to be used to restart the job. If no DSNTABLE DD is present in the JCL, a FATSW15 message will be issued to the operator console asking whether to end the job. If the operator replies 'YES', FATSCOPY will end when it finishes copying the current data set, but will be unable to create the data set needed to restart the job.

STACKING LIMITATIONS

There is one important limitation that you must be aware of when stacking files on tape with FATSCOPY:

The operating system will not allow two files on the same tape volume to be open at the same time.

If an application program attempts to OPEN two tape files concurrently, and you have used FATSCOPY to place both of those files on the same tape, the program will be unable to execute. If this should occur, the first OPEN will be successful but any subsequent OPEN for another file on the same tape will ABEND.

If two separate programs require tape files that you have placed on the same tape with FATSCOPY, and they happen to execute at the same time, one will work but the other will fail or wait.

Unfortunately, there is no simple way to determine which tape files may have this problem, so FATSCOPY is not able to automatically avoid placing such files on the same tape.

Therefore, it is the responsibility of the FATSCOPY user, through appropriate SELECT and EXCLUDE statements, to insure that tape files that may be needed by a single program or by programs executing concurrently are not placed on the same tape.

FDR TAPES

You can use FATSCOPY to copy backup tapes created by FDR, Innovation's backup software.

If some of the files that meet the data set selection criteria are FDRABR Archive or FDRAPPL Application Backup files, the Archive Control File or Application Control File maintained by FDRABR is updated by FATSCOPY when those files have been copied. If you wish to allow Archive and Application Backup files to be selected for copying, specify ABRARC=YES.

The copying of FDRABR Volume Backup tapes (full and incremental volume backups) is *not* restricted by the ABRARC parameter, because these tapes are simply cataloged in the ABR catalog, a standard z/OS system catalog. When FATSCOPY copies these tapes and updates the ABR catalog, the copied tapes can be used by FDRABR. See the FDR manual for the naming conventions used for FDRABR Volume Backups.

SPECIAL TAPES

Innovation does not recommend using FATSCOPY to copy data sets created by proprietary products that record information about the locations of those files in their own databases because FATSCOPY isn't designed to update the data bases. Data sets created by some of these programs, including CA-SAR, are automatically excluded from selection by FATSCOPY.

Consult IBM DFSMShsm documentation for the format of HSM data set names. If you select files with CATDSN=, FATSCOPY will by default exclude HSM backup and migration data sets based on the default names created by IBM; see the HSMMIGMASK= and HSMBAKMASK= operands for details.

FATSCOPY can be used to copy HSM Migration Level 2 tapes by using the HSMML2 keyword. It can also be used to copy tapes that are not recorded in external databases, such as CA-Endevor and ESP/JSS backup tapes.

FATSCOPY can be used to copy OAM data sets, which does record information in its own database, only if you are doing an image copy. (Otherwise, OAM data sets will automatically be excluded from selection by FATSCOPY.) This image copy must be done to a device with the same Media Type as that of the input tapes. After a volume is copied, an OAM UPDATE VOLUME operator command should be done to mark the volume as not writable . In a non-MTL environment, you should also either update the unit esoteric for the volume to reflect the new device for the volume, or use DB2 SPUFI to update the UNITNAME value with the new esoteric in OAM's DB2 TAPEVOL table entry for that volume. Contact Innovation for further details.

FATSCOPY can also be used to copy Mobius ViewDirect data sets if you catalog the output data sets. ViewDirect for MVS can track the files if they are cataloged.

Some special tapes can be copied by FATSCOPY using an image copy, as described below.

IMAGE COPIES

In most cases, an image copy can be used to copy volumes that are used by products that record file locations in their own databases (such as OAM and UPSTREAM). This can be useful if you are migrating tapes from one tape system to another, and you want to avoid the extra processing needed to update the databases recording the files on those tapes. If the product only records information such as volume serial number, file sequence number, and block location on the tape, then you can do a FATSCOPY image copy for tapes created by that product. However, if it also records and uses device type information for the volume, you should do a FATSCOPY image copy for tapes created by that product only if you are copying to the same device type.

Products that record device type information for their files and whose files are thus not suitable for copying with an image copy (unless you are copying to the same device type) include HSM backups, CA-SAR, CA-Disk, CA-Dispatch, Tivoli Space Manager archive tapes, SAS data sets, and DB2 image copies.

Mobius ViewDirect files should not be copied with an image copy if you are copying from one tape model to another with differing physical characteristics; positioning errors reading reports may result. Instead, you should copy and catalog the files with a non-image copy.

When an image copy of volume *vvvvvv* is done to scratch volume *ssssss*, the volume label of the output volume is replaced with *vvvvvv*. Volume *ssssss* no longer exists, but that VOLSER is not deleted from tape management. If you do not want to retain the tape management information for the output volume, you should delete it from your tape management system. The volser of the scratch tape that was overwritten can be determined from the "O/P VOL" column in the Copy Report.

FATSCOPY can be used to write image copies to the following output devices:

- EMC DLm
- EMC MDL
- Stand-alone tape, including all Sun/STK devices and IBM tapes. (Note that the external label for any tape copied to a stand-alone drive needs to be changed after the image copy to make it match the VOLSER of the input tape.)

FATSCOPY cannot be used to write image copies to the following output devices:

- Any IBM TS77xx device
- CopyCross device
- IBM VTF device
- Sun/STK VSM2, VSM3, VSM4, and VSM5 devices.

FATSCOPY will automatically detect any of these devices and not allow them to be used as output for image copies. Support for some of these devices is planned for a future release.

For the following output devices, FATSCOPY can be used to write image copies, but additional steps are required for the copied tapes to be used after the image copy completes:

- Sun/STK 9740
- Any tape library where tapes are mounted by a robot based on the external VOLSER of the tape.

For those devices, the following procedure must be used:

1. Run a FATSCOPY image copy calling for a scratch tape as output.
2. Use the FATSCOPY Copy Report to find the VOLSERs of the input and output volumes.
3. Eject both tapes from the library.
4. Replace the external label of the output tape with an external label that matches the VOLSER of the input tape.
5. Put the output tape back into the tape library.
6. Put a new external label on the input tape, put it back into the library, and run a FATS LABEL job to replace the internal label on the volume so that it matches the new external label.

If you have any questions about whether a device type not listed above is supported for image copy with FATSCOPY, please contact Innovation Technical Support for assistance.

If you are doing an image copy with SMS-managed volumes, you may need to update your ACS routines after the copy. FATSCOPY will update the Storage Group for the volume to be the storage group value for the output device. If your ACS routines use criteria other than the Storage Group to assign the device used to satisfy a tape mount, your ACS routines will need to be updated so that a subsequent job's mount request for the copied volume will be directed to the correct device.

CA 1 CONSIDERATIONS

If your tape management system is CA Technologies' CA 1 (also called TMS), the FATSCOPY support for selecting based on tape management criteria, scratching copied tapes, and updating tape management information is supported when you install the FATSCOPY CA 1 interface using the FATZAPOP program as described in Section 90.3. FATSCOPY supports CA 1 level 5.4 and higher.

If TMSDATA=COPY is specified or defaulted, these fields will be copied from the CA 1 records of the input tape file to the output file:

- creation date and time
- creating job name, step name, and program name
- last used date and time
- last using job name and program name
- expiration date, unless EXPDT= or RETPD= is specified on the TAPEOUT DD statement
- accounting information

If TMSINPUT=SCRATCH is specified, then FATSCOPY will update the expiration date of every successfully copied tape file to the current date + 4. The expiration date of the FIRST data set on the input tape will be set to the highest expiration date of any data set on the tape (or multi-volume tape set). If all data sets on the tape or tape set have been copied by FATSCOPY, FATSCOPY will set the expiration date for the volume to the current date + 4, unless you specified a value other than 4 using the ADDXDAYS keyword.

If you are converting tapes from RMM control to CA 1 control, contact Innovation for information on correctly migrating tape management information between the two systems.

RMM CONSIDERATIONS

Innovation suggests applying the latest maintenance available for your level of RMM. FATSCOPY supports all levels of RMM. You must also install the FATSCOPY RMM interface using the FATZAPOP program as described in Section 90.3.

If TMSDATA=COPY is specified or defaulted, these fields will be copied from the RMM records of the input tape file to the output file:

- creation date and time
- creating job name and program name
- last using job name, step name, and program name
- last read date and time
- last write date and time
- expiration date (of the associated tape volume) unless EXPDT= or RETPD= is specified on the TAPEOUT DD statement
- creating step and DD names
- accounting information

The creation date and creation time of the first data set copied will be assigned as the RMM volume-level "assigned date" and "assigned time" for all output data sets created by FATSCOPY.

If TMSINPUT=SCRATCH is specified, FATSCOPY will change the expiration date for each input data set after it is copied. FATSCOPY will scratch the input tape only if all files on an input tape are being copied *in this FATSCOPY execution* and are eligible to be scratched.

TLMS CONSIDERATIONS

If your tape management system is CA Technologies' TLMS, you must install the FATSCOPY TLMS interface using the FATZAPOP program, as described in Section 90.3. TLMS version 11 and above is supported.

On the TAPEOUT DD statement, you must use DISP=(NEW,KEEP) (or DISP=(OLD,KEEP) when appending to a volume)**.** If KEEP is omitted, FATSCOPY will terminate with an error message.

If TMSDATA=COPY is specified or defaulted, these fields will be copied from the TLMS records of the input tape file to the output file:

- creation date and time
- creating job name, step name, DD name, and program name
- last used date and time
- last using job name
- expiration and keep dates, unless EXPDT= or RETPD= is specified on the TAPEOUT DD statement
- accounting information

When an output volume is created, its control data set number is the default (0, which acts as "1"), even when doing an ALLDSN copy of a volume with a different control data set number. **This may result in the retention of the output volume being different from the retention of the input volume.** If you want to set the control data set number of an output volume to a non-default value, you must use the TLMS UPV command to reset it.

When TMSINPUT=SCRATCH is specified and the control data set is copied, that input file's keep date is set to the highest keep date (or special expiration date, that always count as "higher" than specific Julian keep dates) of all the data sets on the input volume (or multivolume set), rather than the usual scratch date (current date + ADDXDAYS). This is to prevent other data sets from being scratched prematurely. **This may have unexpected results**, such as causing the control data set and volume to be retained **longer** than if TMSINPUT=SCRATCH had not been specified. However, if FATSCOPY successfully copies all the data sets on the volume (or multivolume set) *in this FATSCOPY execution*, then the keep date of the control data set is set to the usual scratch date.

When a data set with keyword expiration "MSG/nnn" is copied, the output data set will have the same expiration. However, **the keep date for the output data set may be different from the input's keep date**. The keep date will be calculated by your installation's TLMS rules. "MSG/nnn" is an exception to the normal TLMS rule that a keyword expiration overrides the keep date when determining volume retention.

A data set with keyword expiration "STATS/nnn" not be selected for copying when SELECT CATDSN is used. When a data set with keyword expiration "STATS/nnn" is copied with SELECT ALLDSN, the copied data set's expiration and keep dates are set to "current date + 2".

Data sets with keyword expiration "FOREIGN" will not be selected for copying.

SMS-MANAGED TAPES

If you want your output tape to be SMS-managed, you can specify STORCLAS= on the TAPEOUT DD statement, or you can specify a data set name on the TAPEOUT DD that is assigned to a SMS-managed tape by your SMS ACS routines. This will cause an SMS-managed tape to be mounted for output. The SMS classes specified or assigned will be used for all data sets copied. However, FATSCOPY does not copy any SMS class information from the input data sets to the output data sets.

If you are doing an image copy with SMS-managed volumes, you may need to update your ACS routines after the copy. FATSCOPY will update the Storage Group of the copied volume to the Storage Group value for the output device. If your ACS routines rely on criteria other than the

Storage Group to assign the device for a tape mount, you will need to update those routines so that volumes which were mounted on the input device before the image copy will be mounted on the output device by subsequent jobs after the image copy.

CONCURRENT COPY JOBS

The only limit to the number of FATSCOPY jobs that you can run at the same time is the number of tape units you have available. To run concurrent jobs, take the following into account:

- Due to z/OS restrictions, each concurrent job's TAPEOUT DD statement must use a different dummy name as its DSN= parameter.
- If you are using an AUDIT DD statement to create audit records, concurrent jobs must use separate data sets. (The FATAUDIT utility can produce a single report using separate audit data sets.)
- Concurrent copy jobs can't read from the same volume at the same time.

WRITING AUDIT RECORDS

FATSCOPY can write an audit record for each data set copied. The information recorded includes the volumes copied, whether the input data set was cataloged, the expiration date of the input data set, the output data sets and volumes created, the expiration date assigned to the output, the number of bytes read and written, and the return code for the copy.

A FATSCOPY job can use one of these two methods for recording audit records:

1. In a sequential data set specified by the AUDIT DD statement. The audit data set can be a disk or tape data set with a disposition of MOD, NEW, SHR, or OLD. We recommend a disposition of MOD to create a single repository for all the audit records.

   If you want to run multiple FATSCOPY jobs simultaneously, you can use a disposition of NEW to create a separate audit file (**each with a different DSN**) in each job. You can then concatenate the output data sets as input to a FATAUDIT job to produce a single report.

   While you can code DISP=SHR or DISP=OLD on the AUDIT DD when using an existing audit data set, **Innovation strongly recommends using MOD or NEW. Using SHR or OLD will wipe out any records that exist in the data set.**

2. In a system logger log stream. This allows you to run concurrent FATSCOPY jobs that write to a single log stream. The z/OS system logger is a component of z/OS that must be activated and configured. You may use either a DASD-only log stream (which is not shared between LPARs) or the duplex coupling facility (to share the log stream between LPARs). For more information on the z/OS system logger see the following IBM publications:

   - *z/OS MVS Setting Up A Sysplex*
   - *z/OS MVS Initialization and Tuning Reference*

   Using the system logger is not recommended unless your installation has experience with using the system logger for other purposes, and is familiar with how to use it.

   You must first perform the following tasks in order to set up the FATSCOPY System Logger environment. Your z/OS systems programmer must do some of these tasks.

   a. Make sure you have authorization to the z/OS system logger address space and that the z/OS system logger (IXGLOGR) is running before you define and use a FATSCOPY log stream. The section "Define Authorization to System Logger address space" in the *z/OS MVS Setting Up a Sysplex* manual provides more information about this.

b. You must also have authorization to the MVS IXCMIAPU utility. This utility is used to define, update, and delete entries in the LOGR couple data set. IXCMIAPU is documented in Appendix B of *z/OS MVS Setting Up a Sysplex*.

c. Use the IXCL1DSU utility to define and format the LOGR couple data set. Your installation may have already done this if you are using the z/OS system logger for other products. This utility is documented in *z/OS MVS Setting Up a Sysplex*.

d. Define the FATSCOPY log stream in the LOGR couple data set using the z/OS utility IXCMIAPU. See the section "Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy" in *z/OS MVS Setting Up a Sysplex*. The following JCL is an example of how to set the values required for a DASD-only FATSCOPY log stream (not shared between LPARs) using the IXCMIAPU utility:

```
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
     DATA TYPE(LOGR) REPORT(YES)
     DEFINE LOGSTREAM NAME(FATSCOPY.AUDITLOG)
            DESCRIPTION(FATSCOPY)
            DASDONLY(YES)
            STG_STORCLAS(TMPDATA)
            LS_STORCLAS(TMPDATA)
            STG_SIZE(48000)
            LS_SIZE(48000)
            HLQ(SYSPXCF)
            LOWOFFLOAD(0)
            HIGHOFFLOAD(50)
            DIAG(YES)
/*
```

For FATSCOPY to use this log stream to write records, use AUDITLOG=*logstreamname* on the COPY statement of each job. Alternatively, if you use the FATZAPOP program to set this value in the FATSCOPY Global Options Table on your system (see section 90.4.1), then by default all copy jobs will use this log stream to write audit records without the need to specify AUDITLOG= in each job.

If you use both an AUDIT DD statement and the AUDITLOG= keyword in the same job, the AUDITLOG= is ignored; audit records will be written to the data set specified in the AUDIT DD statement.

The recorded audit information (sequential data set or log stream) is used as input to the FATAUDIT program. FATAUDIT produces a Detail Report showing all of the data sets copied and a Summary Report showing the total number of files copied, jobs run, etc. See Section 35 for information on using FATAUDIT to produce audit reports.

Although the use of an AUDIT DD or AUDITLOG= is optional, Innovation strongly recommends recording audit information in every FATSCOPY job. Innovation recommends that customers use sequential data sets for recording audit records, unless they have experience using the z/OS System Logger facilities. Using the System Logger in conjunction with setting an AUDITLOG= in your FATSCOPY Options Table (using the FATZAPOP utility) has the advantage that all copy jobs in your shop will have audit information recorded even if your application programmers do not remember to specify audit recording parameters.

EMC DLm CONSIDERATIONS

FATSCOPY can write image copies to the EMC DLm virtual tape system. An image copy replaces the output scratch volume's VOLSER with the input volume's VOLSER. The DLm can recognize that the output VOLSER has been changed so that the new VOLSER can be recognized for subsequent tape mounts. Image copy is described on page 32-31.

When an input volume is scratched on a DLm, the volume is scratched in tape management records but still occupies disk space on the DLm. EMC provides a utility to reclaim this disk space. FDREPORT, another product from Innovation Data Processing, can scan through the tape management data base, select the volumes which have been scratched in tape management, and build a job which you can run to scratch the backing disk space in the DLm. FDREPORT can filter the scratch volumes to select only those meeting criteria that you specify – expiration date, VOLSER mask or range, etc.

Here is an example of an FDREPORT job to generate the DLm scratch job with RMM :

```
//STEP1    EXEC  PGM=FDREPORT,REGION=0M
//MASK      DD *
RMM DV <TVVOLSER> FORCE <TVF1DSN>
/*
//SYSPRINT    DD SYSOUT=*
//SYSPUNCH    DD SYSOUT=*
//SYSABEND    DD SYSOUT=*
//ABRMAP      DD SYSOUT=*
//SYSIN       DD *
**********************
* EXTRACT A WORKING RMM FILE
**********************
EXTRACT PROD=RMM
DEFAULT DISABLE=DUPDSNCHECK
PUNCH   FDRLIB=MASK,ECHO
**********************
* SIMPLE SCRATCH RUN -- ALL SCRATCH VOLUMES WITH & WITHOUT TVF1DSN
**********************
SELECT  TVSTATUS=SCRATCH
PRINT   RPTYPE=SELPCH,DATAT=RMMV,SORT=NO,DISABLE=DUPDSNCHECK
```

This is an excerpt of the output from the FDREPORT job that can be used as input to the DLm scratch processing job:

```
RMM DV AL0C60 FORCE
RMM DV BA0123 FORCE Z.VIDPBK0.B197317A
RMM DV BA1001 FORCE FDRABR.VIDPLB1.B298097A
RMM DV BA1007 FORCE FDRABR.VIDPPM4.B102065A
RMM DV BA1009 FORCE FDRABR.VIDPLB4.B204280A
RMM DV BFL504 FORCE BFL504.FLAT
RMM DV BFL601 FORCE BFL601.FLAT
RMM DV BU0010 FORCE USTPROD.PROD.SERVERS.RETAIN.G0207V00
RMM DV BU0112 FORCE
RMM DV BU0113 FORCE
RMM DV BU1001 FORCE USTPROD.ALEXSU10.COPYF2.G0142V00
…
```

You can find out more information on FDREPORT by visiting Innovation's web site:

http://www.fdr.com/products/fdreport/

33.0 FATSCOPY EXAMPLES

EXAMPLE 1: MIGRATE CERTAIN DATA SETS TO AN IBM 3590

Cataloged data sets matching the data set name mask will be copied and stacked on an IBM 3590 cartridge. Only data sets that have expiration dates of January 1, 2012 or later will be copied. (Data sets that have earlier expiration dates will not be copied.) The copied files will be recataloged to the output tape, and the expiration dates for the input data sets will be set to the current date+4. The output data sets will be stacked on a single output cartridge regardless of their expiration date (since EXPDTGROUP=999 has been used), unless more than one cartridge is required to hold all the data sets. Since this is a migration, multi-file input tapes will be copied.

```
//STACK    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD DSN=OUTPUT,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
   COPY    TMSINPUT=SCRATCH,EXPDTGROUP=999,MULTIFILE=YES
   SELECT  CATDSN=TECH%.*.SAVE.**,EXPDATE>=2012001
```

EXAMPLE 2: STACK FILES CATALOGED TO 3490E ON STORAGETEK T9840

Data sets created in 2010 and cataloged to the 3490E tape device type will be copied and stacked on StorageTek T9840 cartridges (esoteric name STK9840 has been defined in this installation to allocate a T9840 drive). If more than 500 files are selected, a fresh scratch tape will be mounted for each set of 500 output files. The copied files will be recataloged to the output tape, and the input tapes will be retained by your tape management system; you may wish to scratch them or delete them from tape management control later. Certain data sets (system data sets and ABR Volume Backups) are excluded.

Data sets will be grouped by their expiration date; data sets that expire within a range of 7 days will be stacked together on one cartridge so that all files on the cartridge will expire at about the same time. Data sets with special expirations, such as 99000 for catalog control, will be stacked on a separate cartridge.

```
//STACK    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD DSN=OUTPUT,UNIT=STK9840,DISP=(,KEEP)
//SYSIN    DD *
   COPY    MAXFILE=500
   EXCLUDE CATDSN=SYS**
   EXCLUDE CATDSN=FDRABR.**
   SELECT  CATDSN=**,DEVTYPE=3490,CRDATE>=2010001,CRDATE<=2010365
```

EXAMPLE 3: SIMULATE STACKING OF DATA SETS

Data set stacking will be simulated. FATSCOPY will select cataloged data sets matching the CATDSN= mask that were created by jobs beginning with PAY or INV and the listing will show all selected data sets. Since this may select data sets created by FDR Application Backup (FDRAPPL), an EXCLUDE statement for FDRAPPL data sets is provided.

```
//SIMULATE EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD DUMMY
//SYSIN    DD *
   SIM
   EXCLUDE CATDSN=PROD*.V//////.B++++++/
   SELECT  CATDSN=PROD**,CRJOB=(PAY*,INV*)
```

EXAMPLE 4: COPY TAPE VOLUMES INTO A VTS

All files on 3 selected input volumes will be copied into a VTS (virtual tape system) as specified by UNIT=VTS on the TAPEOUT statement. The 3 tape volumes will be identified from tape management information and all files on those tapes will be copied. Each input tape will be copied to a separate VTS virtual tape. If any of the tapes are part of a multi-volume tape set, then all files on all tapes in the set will be copied. Output files will be re-cataloged to the output tapes only if they were cataloged to the input tapes (CAT=ONLY is the default).

```
//VTS      EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD DSN=OUTPUT,UNIT=VTS,DISP=(,KEEP)
//SYSIN    DD *
   COPY    CAT=ONLY
   SELECT  ALLDSN,VOL=030427
   SELECT  ALLDSN,VOL=010034
   SELECT  ALLDSN,VOL=004961
```

EXAMPLE 5: STACK TAPE VOLUMES ONTO AN IBM 3590

All files on 5 selected input volumes will be stacked on a single volume on an IBM TS1130 tape drive. The 5 volumes will be identified from tape management information and all files on those tapes will be copied and stacked on the output. If any of the tapes are part of a multi-volume tape set, then all files on all tapes in the set will be copied. EXPDTGROUP=999 instructs FATSCOPY to stack all the data sets together regardless of their expiration dates. All copied files will be cataloged to the output tape, regardless of the catalog status of the input files.

```
//STACK    EXEC  PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD DSN=OUTPUT,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
   COPY    EXPDTGROUP=999,CAT=RECAT
   SELECT  ALLDSN,VOL=A03452
   SELECT  ALLDSN,VOL=C19498
   SELECT  ALLDSN,VOL=C19502
   SELECT  ALLDSN,VOL=R32343
   SELECT  ALLDSN,VOL=F00042
```

EXAMPLE 6: ALLOCATE 20 BUFFERS AND STACK OUTPUT DATA SETS

Cataloged data sets that match the data set and volume masks will be copied to an IBM 3592 tape. All copied files will be cataloged to the output tape, regardless of the catalog status of the input files. FATSCOPY will allocate 20 buffers and create 2 CCW chains that will each read 10 blocks at a time.

```
//STACK    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//TAPEOUT  DD DSN=DUMMY,UNIT=3592,LABEL=(,SL)
//SYSIN    DD *
 COPY   CAT=RECAT,BUFNO=20
 SELECT CATDSN=MAGSTAR.**,VOL=003*
 SELECT CATDSN=T99000.CPUC.**,VOL=003*
 SELECT CATDSN=TEST256K.**,VOL=003*
 SELECT CATDSN=TESTVT90.**,VOL=003*
/*
```

EXAMPLE 7: CREATE A CHECKPOINT DATA SET

FATSCOPY will do a simulation run, selecting data sets with the high level qualifier of FATSTEST from the system catalog and creating a checkpoint data set on disk that will be used as input to a second job that will copy all the data sets selected. Note that SPACE, UNIT, and DISP=(,CATLG) are used to allocate a new DSNTABLE data set.

```
//STEP1    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//TAPEOUT  DD DUMMY
//DSNTABLE DD DSN=FATSCOPY.CHKPOINT,SPACE=(TRK,(15,5),RLSE),
//            UNIT=SYSALLDA,DISP=(,CATLG)
//SYSIN    DD *
 SIM    CHECKPT
 SELECT CATDSN=FATSTEST.**
/*
```

EXAMPLE 8: COPY DATA SETS USING A CHECKPOINT DATA SET

Run a job to copy data sets selected by the job in the prior example. The data sets will be copied to a 3590 tape. All of the output data sets will be recataloged. Note that DISP=SHR is used for the existing DSNTABLE data set. If this job is interrupted, this will allow the DSNTABLE data set to be rewritten for a subsequent RESTART job.

```
//STEP1    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//DSNTABLE DD DSN=FATSCOPY.CHKPOINT,DISP=SHR
//TAPEOUT  DD DSN=DUMMY,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
  COPY MULTIFILE=YES,CAT=RECAT,RESTART
/*
```

EXAMPLE 9: CREATE A TAPE WHICH ALLOWS DATA SETS TO BE ADDED TO THE END OF THE TAPE

This is run using the LASTAPE keyword using a data set name, FATSCOPY.LASTAPE.DSN, that is not already cataloged. A scratch tape will be mounted; FATSCOPY will copy the selected data sets to the tape, and then create and catalog the empty data set FATSCOPY.LASTAPE.DSN at the end of the output tape as a last-file-on-tape marker.

```
//STEP1    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//TAPEOUT  DD DSN=DUMMY,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
   COPY   MULTIFILE=YES,LASTAPE=FATSCOPY.LASTAPE.DSN
   SELECT CATDSN=FATSTEST.SOMEFILE.**
/*
```

EXAMPLE 10: ADD DATA SETS TO THE END OF AN EXISTING TAPE

This is run using the name of an existing empty data set, FATSCOPY.LASTAPE.DSN, that is already cataloged, and is used as the last-file-on-tape marker. The volume containing that data set will be mounted and repositioned to the location of FATSCOPY.LASTAPE.DSN. FATSCOPY then will copy the selected data sets to the tape, and re-create and re-catalog the data set FATSCOPY.LASTAPE.DSN at the new end of the output tape. THRESHOLD=95 tells FATSCOPY to not add any more data sets to the tape once it is at least 95% full; in that case FATSCOPY will call for a scratch tape to continue copying files.

```
//STEP2    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//TAPEOUT  DD DSN=DUMMY,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
   COPY   MULTIFILE=YES,LASTAPE=FATSCOPY.LASTAPE.DSN,THRESHOLD=95
   SELECT CATDSN=FATSTEST.MOREFILE.**
/*
```

EXAMPLE 11: COPY WITH RENAME

This will copy all cataloged data sets whose names match the mask JAT.EE.FATS.*, change the second index level for each output data set from “EE” to “TEMP”, and catalog the output data sets with their new names (assuming those names are not already cataloged). SHOWNEWN instructs FATSCOPY to display the new output data set names in the Copy Report. EXPDTGROUP=999 instructs FATSCOPY to stack all the data sets together regardless of their expiration dates. The input files are left unchanged and remain cataloged.

```
//STEP1    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//TAPEOUT  DD DSN=DUMMY,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
       COPY   MULTIFILE=YES,CAT=YES,SHOWNEWN,EXPDTGROUP=999
       SELECT CATDSN=JAT.EE.FATS.*
       RENAME LF=ALL,NEWI=.TEMP
/*
```

EXAMPLE 12: COPY WITH RENAME FOR MIGRATION

A customer wants to copy all the files with the high level qualifier JULY from volumes V00100, V00101, and V00102 to another tape(s) for migrating to a new site, leaving the original files cataloged. The output files will be renamed by changing their high-level qualifiers to the temporary value XULY, and will be cataloged. (At the new site the process can be reversed to copy the XULY files, renaming them back to JULY and cataloging them.) SHOWNEWN instructs FATSCOPY to display the new output data set names in the Copy Report. EXPDTGROUP=999 instructs FATSCOPY to stack all the data sets together regardless of their expiration dates.

```
//STEP1    EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//TAPEOUT  DD DSN=DUMMY,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD *
   COPY   MULTIFILE=YES,CAT=YES,SHOWNEWN,EXPDTGROUP=999
   SELECT CATDSN=JULY.**,VOL=(V00100,V00101,V00102)
   RENAME LF=ALL,NEWI=XULY
/*
```

EXAMPLE 13: SIMULATE COPYING PHYSICAL VOLUMES FROM AN IBM TS7700 VIRTUAL TAPE SYSTEM

FATSCOPY will do a simulation, selecting all data sets on all virtual volumes located on physical volumes TS0123 and TS0999 in an IBM TS7700 virtual tape system. The MAPTAPE DD statement is required for FATSCOPY to obtain physical-to-virtual volume mapping information from the input virtual tape device (VTSIN). (To actually copy the data sets, change SIM to COPY, and un-comment the TAPEOUT DD statement.)

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//* TAPEOUT DD UNIT=outputdevice,DSN=DUMMY,DISP=(,KEEP)
//MAPTAPE  DD UNIT=VTSIN,DISP=(,CATLG),DSN=JAT.MAPTAP.FILE,RETPD=1,
//            LABEL=(,SL),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,TRTCH=NOCOMP)
//SYSIN    DD *
 SIM    MULTIFILE=YES,VIRTTYPE=IBM
 SELECT ALLDSN,PHYSVOL=TS0123
 SELECT ALLDSN,PHYSVOL=TS0999
/*
```

EXAMPLE 14: COPY PHYSICAL VOLUMES FROM AN STK VSM VIRTUAL TAPE SYSTEM

FATSCOPY will do a copy, selecting all data sets on all virtual volumes located on physical volumes STK123 and STK999 in a StorageTek VSM virtual tape system, copying them to the tape unit specified on the TAPEOUT DD statement. The STEPLIB DD statement for your HSC link library is required, unless this library is in the linklist or on the JOBLIB statement. The DSNTABLE statement is used so that the copy job can be restarted (by another job using the RESTART keyword) if you need to interrupt this job with a STOP or CANCEL command. The RESTART job would continue copying files starting where the interrupted job stopped.

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB  DD DSN=your.sls.prod.linklib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//DSNTABLE DD DSN=FATSCOPY.MIG.CHKPOINT,SPACE=(TRK,(15,5),RLSE),
//            UNIT=SYSALLDA,DISP=(,CATLG)
//TAPEOUT  DD UNIT=outputdevice,DSN=DUMMY,DISP=(,KEEP)
//SYSIN    DD *
 COPY   MULTIFILE=YES,VIRTTYPE=STK
 SELECT ALLDSN,PHYSVOL=STK123
 SELECT ALLDSN,PHYSVOL=STK999
/*
```

EXAMPLE 15: USE A LOGICAL VOLSER TO COPY AN ENTIRE PHYSICAL VOLUME ON AN STK VSM VIRTUAL TAPE SYSTEM

FATSCOPY will do a copy, selecting all data sets on all virtual volumes that are located on the same physical volume as logical volume 345123, copying them to the tape unit specified in the TAPEOUT DD statement. The STEPLIB DD statement for your HSC link library is required, unless this library is in the linklist or on the JOBLIB statement. The DSNTABLE statement is used so that the copy job can be restarted (by another job using the RESTART keyword) if you need to interrupt this job with a STOP or CANCEL command. (The RESTART job would continue copying files starting where the interrupted job stopped.)

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB  DD DSN=your.sls.prod.linklib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//DSNTABLE DD DSN=FATSCOPY.MIG.CHKPOINT,SPACE=(TRK,(15,5),RLSE),
//            UNIT=SYSALLDA,DISP=(,CATLG)
//TAPEOUT  DD UNIT=outputdevice,DSN=DUMMY,DISP=(,KEEP)
//SYSIN    DD *
 COPY   MULTIFILE=YES,VIRTTYPE=STK
 SELECT ALLDSN,ALLPHYS=345123
/*
```

EXAMPLE 16: MAKING AN IMAGE COPY

To make an exact bit-for-bit copy of 5 input tapes onto 5 output tapes (as you might want to do when migrating from one tape device to another), a FATSCOPY image copy job is used. No stacking of files is done; each output volume contains the same files as the corresponding input volume. To your tape management system and application programs, the copied tapes look exactly like they did before they were copied, except that each is recorded by tape management as being located on the new device. If any of the specified volumes is a part of a multivolume set, an image copy will be done for each volume of that set.

Please note the restrictions on page 32-31 for using IMAGE.

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD UNIT=newdevice,DSN=DUMMY,DISP=(,KEEP)
//SYSIN    DD *
 COPY   IMAGE
 SELECT ALLDSN,VOL=100201
 SELECT ALLDSN,VOL=100502
 SELECT ALLDSN,VOL=200203
 SELECT ALLDSN,VOL=100504
 SELECT ALLDSN,VOL=740205
/*
```

EXAMPLE 17: IMAGE COPY FOR A RANGE OF VOLUMES

To make an exact bit-for-bit copy of 5 input tapes with consecutive volume serial numbers onto 5 output tapes (as you might want to do when migrating from one tape device to another), a FATSCOPY image copy job is used. No stacking of files is done; each output volume contains the same files as the corresponding input volume. To your tape management system and application programs, the copied tapes look exactly like they did before they were copied, except that each is recorded by tape management as being located on the new device.

This job will make an image copy of volumes 100201, 100202, 100203, 100204, and 100205. If any of the specified volumes is a part of a multivolume set, an image copy will be done for each volume of that set. If any data set on one of these volumes was last used by FATSCOPY, the EXLASTPGM= parameter ensures that entire volume will be bypassed. This can be used to avoid copying volumes that were already copied by another job, and is useful when a volume may be part of a multivolume set.

Please note the restrictions on page 32-31 for using IMAGE.

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT   DD SYSOUT=*
//COPYRPT  DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT  DD UNIT=newdevice,DSN=DUMMY,DISP=(,KEEP)
//SYSIN    DD *
 COPY   IMAGE,EXLASTPGM=FATSCOPY
 SELECT ALLDSN,VOL=100201,NUMVOLS=5
/*
```

EXAMPLE 18: MIGRATING UPSTREAM TAPES BETWEEN DEVICE TYPES

Users of Innovation's Upstream product who want to migrate their z/OS Upstream tapes between devices can use the following procedure:

1. Obtain a list of dataset names and volume serials numbers for the VAULT Retention datasets on your VAULT tapes. The following JCL can be used to do this (customize the LVL parameter as necessary):

```
//COPYCAT  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
  LISTCAT LVL(USTPROD.RETAIN) VOLUMES
/*
```

2. Take that list of volume serial numbers and run the following COPY job, specifying the first VOLSER for each set of VAULT Retention files on a different SELECT control statement:

```
//COPYCAT  EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//SELRPT    DD  SYSOUT=*
//COPYRPT   DD  SYSOUT=*
//SYSABEND  DD  SYSOUT=*
//ERRORRPT  DD  SYSOUT=*
//TAPEOUT   DD DSN=DUMMY,UNIT=tapedevice,DISP=(,KEEP)
//SYSIN     DD  *
  COPY    CAT=RECAT
  SELECT  ALLDSN,VOL=aaaaaa
  SELECT  ALLDSN,VOL=bbbbbb
  …
/*
```

3. If you intend to utilize the backups to perform restores or to be input to future full merge backups, the new backup datasets need to be processed by the Upstream zOS REGEN utility. This will load the proper new VOLSER and client file tape block offsets into the Upstream zOS FILEINFO file. If you do not plan to immediately use these tapes, you may defer this step until immediately prior to performing an Upstream function that would use the new tapes as input. If the retention of these newly created tape volumes expires prior to you needing to use them, then all this processing overhead of the REGEN can be avoided.
4. Note: if you are copying into an EMC DLm virtual tape system, you can use IMAGE in step #2; in that case the REGEN step is not necessary.

EXAMPLE 18: FDREPORT JOB TO LIST VOLUMES GROUPPED BY CREATING JOB

This example shows how the companion product FDREPORT can be used to list the tapes on your system, grouped by creating job, and giving a total of the number of tapes in each group.

```
//FDREPORT EXEC PGM=FDREPORT,REGION=0M
//RMMXTR   DD  DISP=SHR,DSN=your.fdreport.extract.file
//SYSPRINT DD  SYSOUT=*
//ABRMAP   DD  SYSOUT=*
//SYSIN    DD  *
  TITLE   LINE='LIST AND GROUPED PER CREATING JOB - <TVCRJOB> +
                -  EXTRACT FILE <TVXTDATE>'
  XEXCLUDE TVCRJOB=' '
  XSELECT  TVVOLSER=*
  REPORT FIELDS=(TVVOLSER,TVCRJOB,TVEXDATE,TVUSEBYT)
  SORT FIELD=(TVCRJOB,TVVOLSER),BREAK=(SEJ,NO)
  SUMMARY FIELD=(TVVOLSER,TVUSEBYT)
  PRINT DATATYPE=RMMVOL,DISABLE=SUMVALUES,BYTEFORMAT=MEGABYTES,
                SORTALLOC=CYL,SORTCORE=8000000
/*
```

Here is a sample output:

```
          LIST AND GROUPED PER CREATING JOB - DBJK201M -  EXTRACT FILE 2013.326

VOLSER CREATJOB   EXPDAT VOLUSEBYTES
------ -------- -------- -----------
251218 DBJK201M 2013.313      0.029M
251243 DBJK201M 2013.313      0.057M
251245 DBJK201M 2013.313      0.029M
……
251393 DBJK201M 2013.313      0.029M
251399 DBJK201M 2013.313      0.057M
    SUBTOTAL -- TVCRJOB---DBJK201M
      TVVOLSER----------22  TVUSEBYT-936742.912K

          LIST AND GROUPED PER CREATING JOB - DFSMSHSM -  EXTRACT FILE 2013.326
VOLSER CREATJOB   VEXDAT VOLUSEBYTES
------ -------- -------- -----------
250204 DFSMSHSM 1999.365    714.195M
250348 DFSMSHSM 2013.331    662.310M
……
253531 DFSMSHSM 1999.365    185.418M
    SUBTOTAL -- TDCRJOB---DFSMSHSM
      TDVOLSER----------63  TDAPRSIZ--97032.084M
…
```

EXAMPLE 19: FDREPORT JOB TO LIST VOLUMES REFERENCED MORE THAN 430 DAYS AGO

This example shows how the companion product FDREPORT can be used to list all of the tapes in your system which were last referenced more than 430 days ago.

```
//FDREPORT EXEC PGM=FDREPORT,REGION=0M
//RMMXTR   DD  DISP=SHR,DSN=your.rmm.extract.file
//SYSPRINT DD  SYSOUT=*
//ABRMAP   DD  SYSOUT=*
//SYSIN    DD  *
  XEXCLUDE TVSTATUS=(INIT,SCRATCH)
  XSELECT  TVLRDAYS>430
  TITLE    LINE='TAPE VOLS LAST REFRENCED MORE THAN 430 DAYS AGO +
                - EXTRACT FILE <TVXTDATE>'
  REPORT   FIELD=(TVVOLSER,TVLRDATE,TVCRDATE,
                  TVEXDATE,TVF1DSN)
  SORT     FIELD=(TVLRDATE,TVVOLSER)
  SUMMARY  FIELD=(TVVOLSER,TVUSEBYT)
  PRINT    DATATYPE=RMMVOL,DISABLE=SUMVALUES
/*
```

Here is a sample output:

```
TAPE VOLS LAST REFRENCED MORE THAN 430 DAYS AGO - EXTRACT FILE 2013.326

VOLSER   LRDATE   CRDATE   EXPDAT FIRST FILE DATASET
------ -------- -------- -------- ------------------
300100 2002.065 2002.065 2002.065 D03.A04077.AD9F2F.LCCC0040.SORT0503.VPLINK
300220 2002.067 2002.066 2002.067 D03.A04050.RESATRT.LCJD601M.EQUAL
301518 2002.070 2002.070 2002.070 D03.A04726.TS030.UPLOD01.WZIPS
301639 2002.070 2002.070 2002.070 D06.A04050.RESTART.OFJD397P.SHISTORY.V#1
301640 2002.070 2002.070 2002.070 A03.A04077.LC.LCJC002P.BACKALL.G0422V00
302289 2002.071 2002.071 2029.208 A03.A04234.AHPPMSP.OHCONV.SAVE
301104 2002.075 2002.068 2029.205 A06.A04077.SL.LCJD601P.PROD.YTD.G0170V00
301130 2002.075 2002.068 2029.205 A03.A04077.LC.LCJD001P.PROD.YTD.G0171V00
304319 2002.077 2002.074 2002.077 A03.A04077.LC.LCJC002P.BACKALL.G0423V00
301295 2002.080 2002.068 2099.365 A03.A04345.RSDGR16.C102068.T054336.E2099365
.....
411789 2012.265 2010.119 2012.264 A03.AA0242.TAI.TAJI001P.TAITAIX.G0513V00
428433 2012.265 2010.229 2012.269 A03.A04346.SPFI0072.G2019V00
443895 2012.265 2000.000 2014.033 M03.A04077.LC.LCJC001P.COMM.G0038V00
449236 2012.265 2000.000 2012.142 M06.A04077.SL.LCJD007P.WNBHIST.G0022V00
449443 2012.265 2000.000 2012.142 M30.A04077.NY.LCJD007P.WNBHIST.G0022V00

   FINAL TOTALS --

         TVVOLSER-------69691  TVUSEBYT--67635.361G
```

This page intentionally left blank.

34.0 FATSCOPY PANELS

FATSCOPY panels help you to perform many of the functions performed by batch jobs. These panels can create FATSCOPY JCL and control statements as job streams that can be submitted, edited before submission, or saved for reuse. In addition, they offer online simulations that can be used as input to a later batch job to copy the data sets selected for copying. These panels can simplify the use of FATSCOPY.

In most cases, you advance to the next panel by hitting the END key (usually PF3). Entering CAN takes you back to the first panel in the sequence (not necessarily the immediately preceding panel).

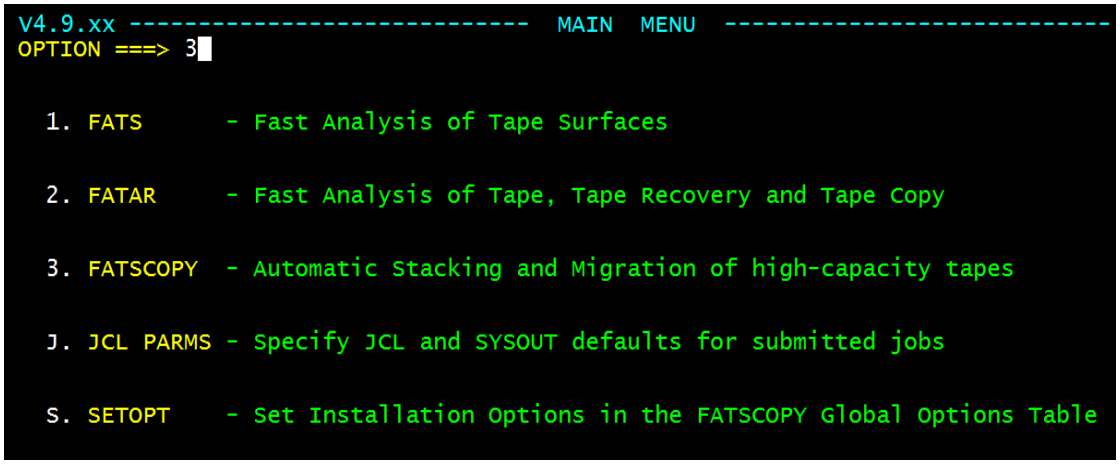
The actual ISPF panels may vary slightly from the examples in this manual.

PANEL HELP INFORMATION

Every panel has detailed HELP information that can be displayed by pressing the HELP key (usually PF1). If your level of ISPF supports it, field-level help for each keyword is available by placing the cursor in the input field in question and pressing HELP.

The descriptions of the fields and functions on the panels in the following sections are brief because the detailed, current descriptions are in the HELP.

MAIN MENU


V4.9.xx ---------- MAIN MENU ----------
OPTION ===> 3
1. FATS - Fast Analysis of Tape Surfaces
2. FATAR - Fast Analysis of Tape, Tape Recovery and Tape Copy
3. FATSCOPY - Automatic Stacking and Migration of high-capacity tapes
J. JCL PARMS - Specify JCL and SYSOUT defaults for submitted jobs
S. SETOPT - Set Installation Options in the FATSCOPY Global Options Table

FATS/FATAR/FATSCOPY Main Menu

This is the FATS/FATAR/FATSCOPY Main Menu. **The remainder of this section of the manual covers only the FATSCOPY panels. The FATS/FATAR panels are covered in Section 93.0.**

Two of the selections on this menu are used to configure FATSCOPY.

- **SETOPT** – This selection is used by the system administrator. **It must be used at least once before any FATSCOPY jobs are run** in order to define which tape management system is being used on your system. It can also be used at any time to change the default values saved for many FATSCOPY keywords. This is described on page 34-2.

- **JCL PARMS** – Each user needs to select this option the first time they use the FATSCOPY panels in order to set JCL defaults in their ISPF user profile. This is described on page 34-7.

If you've already set your JCL defaults with the JCL PARMS panel, you can select Option 3 to proceed directly to the FATSCOPY Selection Menu, shown on page 34-8.

(*Note*: Previous versions of FATSCOPY had Option Q on the Main Menu to query tape management information. This option has been moved to the FATSCOPY Selection Menu.)

UPDATING GLOBAL OPTIONS

```
V4.9.xx     FATSCOPY INSTALLATION -- SET FATSCOPY GLOBAL OPTIONS PRIMARY MENU
OPTION ==>


         1 Global Selection Options

         2 Miscellaneous Options

         3 Copy Options

         4 Data Set Names and Masks

         5 System Logger Log Stream Options

  SAVE - Save option changes            COPY - Copy options from a prior level
CANCEL - Exit without saving changes   AUDIT - Display user changed options
                                       RESET - Re-Initialize all options


FATSCOPY LIBRARY: FATSYS.FAT49XX.LOAD                          <
```

FATSCOPY Global Options Primary Menu

The default values for many FATSCOPY COPY/SIM keywords are stored in the FATSCOPY Global Options Table. Each FATSCOPY job reads this table to determine the current value for these keywords. The Global Options Primary Menu is the starting point for changing or displaying these values. Normally, only the system programmer will have authority to update and save the FATSCOPY Global Options.

From this panel, you can

- Select panels to individually set options (options "1" through "5");
- "COPY" an entire set of options from a previous release of FATSCOPY;
- "RESET" your options values to the original values delivered by Innovation;
- Display options that have been changed at your site using the AUDIT command.

Be sure that the correct FATSCOPY library name is entered at the bottom of the panel.

To individually set options for FATSCOPY keywords, use Options 1 through 5. Each of these will bring you to another panel with a set of FATSCOPY options. These five panels are shown below.

```
V4.9.xx ----------------- FATSCOPY / GLOBAL SELECTION OPTIONS -----------------
COMMAND ===>


ABRARC     - Copy ABR Archive Backups.............................. NO  (Yes,No)
HSMML2     - Copy HSM ML2 vOlumes.................................. NO  (Yes,No)
MULTIFILE  - Copy multifile volumes................................ NO  (Yes,No)
MULTIVOL   - Copy multivolume sets.............................. YES  (Yes,No,Only)
OFFSITE    - Copy offsite volumes.............................. NO   (Yes,No,Only)
SORT       - Sort files in volume and file sequence number......... YES (Yes,No)
MAXTOTFILE- Maximum number of files to select for copying...... 1000  (1-65535)
EXPDTGROUP- Group data sets expiring within x days together........ 7   (0-999)
OPCAP      - Gigabyte capacity of output tapes..................... 10   (1-1000)
SELTERR    - Issue non-zero return code if selection errors occur.. YES (Yes,No)
VIRTTYPE   - VTS manufacturer (used to map backend tapes in a VTS) ___ (IBM/STK)
```

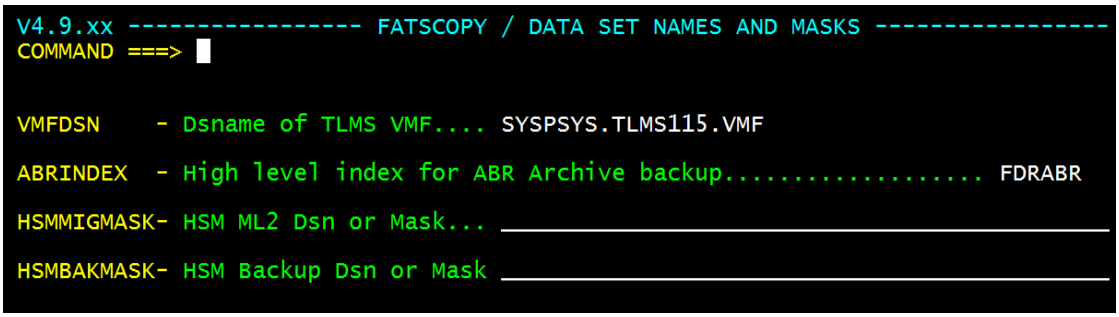
Menu Option 1: Global Selection Options

```
V4.9.xx ------------------- FATSCOPY / MISCELLANEOUS OPTIONS ------------------
COMMAND ===>


TMSIN      - Active Tape Management System................... RMM  (RMM,CA1,TLMS)
LINECNT    - Max number of print lines per page.................. 60    (1,32767)
MAXFILE    - Max files per output ( 0 equals stack until full).. 0     (0-65535)
ABEND      - Abnormal termination Option ................ ABEND   (ABEND,Retcode)
ECHO       - Echo current options to Sysprint..................... NO     (Yes,No)
```

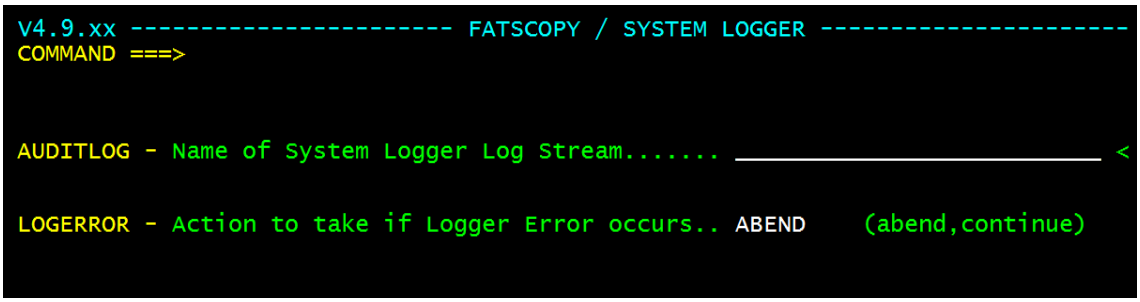
Menu Option 2: Miscellaneous Options

```
V4.9.xx ----------------------- FATSCOPY / COPY OPTIONS -----------------------
COMMAND ===>


BUFNO      - Number of Data Buffers..................................... 10 (2-25)
CAT        - Catalog output data sets.................. ONLY  (Only,Recat,Yes,No)
NOTCATRC   - Retcode issued if cataloging of output data sets fails... 0 (0,4,8)
LASTFILE   - Value to determine whether to rewrite entire file.... 1000 (1-1000)
MAXERR     - Maximum number of permanent data checks per file... 1     (1,32767)
MAJERR     - Maximum number of non-data check errors per file... 1     (1,32767)
MAXTERR    - Maximum number of temporary data checks per file... 100   (1,32767)
DUMP       - Produce dump for U0888, U0300, U0502, and U0913... NO  (Yes,No,Ask)
TMSDATA    - Copy Tape Management Data from Input to Output COPY   (Copy,Nocopy)
TMSINPUT   - Keep or Expire input data sets.............. KEEP    (Keep,Scratch)
ADDXDAYS   - Nbr of days to add to current date when expiring inputs 4   (0-999)
SUPULAB    - Suppress copying user labels to output files........... NO  (Yes,No)
RMMFAIL    - Action to take if RMM Updates fail.. CONTINUE (Continue,Stop,Retry)
RETRY      - Number of times to retry failed RMM updates............. 0   (0-255)
```

Menu Option 3: Copy Options


V4.9.xx ---------- FATSCOPY / DATA SET NAMES AND MASKS ----------
COMMAND ===>
VMFDSN - Dsname of TLMS VMF.... SYSPSYS.TLMS115.VMF
ABRINDEX - High level index for ABR Archive backup..... FDRABR
HSMMIGMASK- HSM ML2 Dsn or Mask...
HSMBAKMASK- HSM Backup Dsn or Mask

Menu Option 4: Data Set Names and Masks


V4.9.xx ---------- FATSCOPY / SYSTEM LOGGER ----------
COMMAND ===>
AUDITLOG - Name of System Logger Log Stream.....
LOGERROR - Action to take if Logger Error occurs.. ABEND (abend,continue)

Menu Option 5: System Logger Log Stream Options

On each panel, the current value in effect for each keyword is shown. Make any changes to the values on the panel you want, then hit END to return to the Global Primary Options Menu with the changed values, or CAN to return without the changed values.

IMPORTANT: While you can update your FATSCOPY defaults at any time, **there is one value which MUST be set before any FATSCOPY jobs are run. On the Menu Option 2 panel, you must set a value for "TMSIN – Active Tape Management System".** If using TLMS, you must also set a value for "VMFDSN – Dsname of TLMS VMF" on the Menu Option 4 panel. (You can skip this if you have already set the tape management information by running a FATZAPOP batch job during product installation with ZAP TMSIN=.)

When you have returned to the Global Options Primary Menu after changing whichever default values you with to alter, enter SAVE on the command line to save all updated values to the FATSCOPY Global Options Table. **If you exit the Primary Menu with the CANCEL command, no changes made on any of the Option 1 through 5 panels will be saved.**

COPY GLOBAL OPTIONS FROM A PREVIOUS RELEASE

If you are upgrading from a previous release of FATSCOPY which supported the Global Options Table (version 4.9.26 or above) and you want to copy all of the default values you saved in that earlier release, you do not need to go through Options 1 through 5. Instead, enter the COPY command on the Global Options Primary Menu to bring up the following panel:

```
V4.9.xx -------------------- FATSCOPY / COPY GLOBAL OPTIONS -------------------
OPTIONS==>
Enter COPY to Copy the Library                      Enter END to Exit (no Copy)

WARNING: PLEASE MAKE SURE THAT THE COPY TO/FROM DATA SETS NAMES ARE CORRECT

COPY "TO" FATSCOPY LIBRARY DATA SET:
     Data Set Name ===> FATSYS.FAT49XX.LOAD                              <

COPY "FROM" FATSCOPY LIBRARY DATA SET:
     Data Set Name ===> _____________________________________________    <
```

Copy Global Options from a Previous Release

This panel allows you to specify the library containing the object modules for the FATSCOPY version you want to copy default values from, and displays the library for the current version of FATSCOPY to be updated. Enter the fully qualified “from” library name (be sure the library is version 4.9.26 or later), and then enter COPY.

This panel can also be used when upgrading from a trial version of FATSCOPY to a production version. Specify the library containing the trial version as the “from” data set.

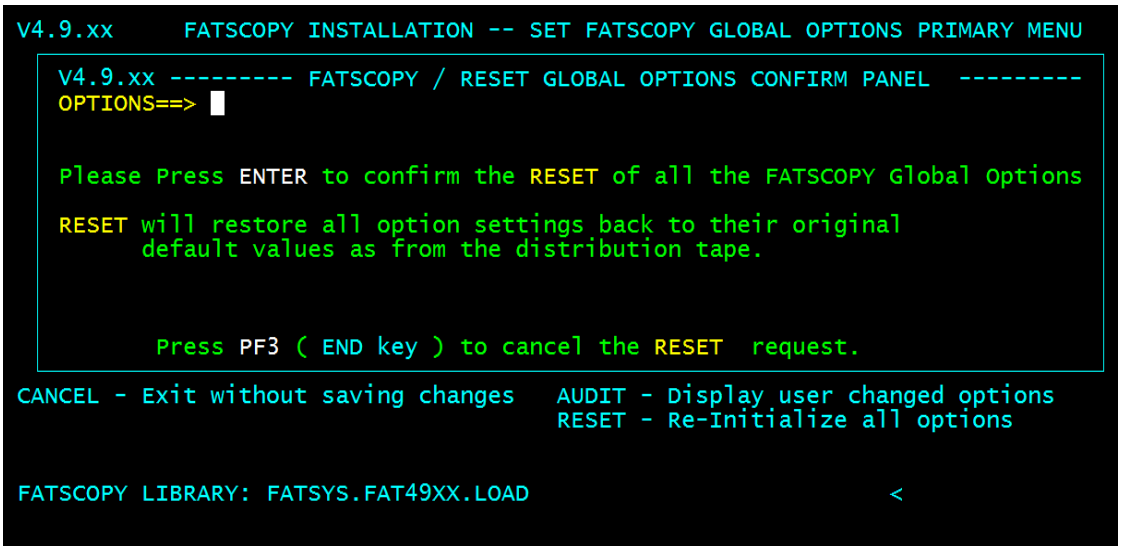
DISPLAY DEFAULTS WHICH HAVE BEEN CHANGED FROM THE DELIVERED DEFAULTS

```
 BROWSE    SYS12278.T170732.RA000.METO.R0217250---------------------------------------- Line 00000000 Col 001 120
 Command ===>                                                                          Scroll ===> PAGE
********************************************** Top of Data ********************************************************
FATS400 FATSCOPY GLOBAL OPTION(S) CHANGE -- FATZAPOP VER 4.9.xxP --INNOVATION DATA PROCESSING  DATE - 12.278  PAGE - 001
FATS303   CARD IMAGE - *  AUDIT                                                                 *
FATS400 FATSCOPY GLOBAL OPTION(S) CHANGE -- FATZAPOP VER 4.9.xxP --INNOVATION DATA PROCESSING  DATE - 12.278  PAGE - 002
FATS549  PRINTING FATOPT   -- SYSLIB=FATSYS.FAT49XX.LOAD

  VERSION -- 4.9.xxP    DATE -- 12.208

VALUES CHANGED IN OPTIONS TABLE

* TMSIN     ACTIVE TAPE MANAGEMENT SYSTEM.............................RMM


* VMFDSN    DSNAME OF TLMS VMF.....SYSPSYS.TLMS115.VMF


FATS499   FATZAPOP(4.9.xxP ) PROCESSING COMPLETED
********************************************** Bottom of Data *****************************************************
```

AUDIT Display

On the Global Options Primary Menu, entering AUDIT will result in the display of a panel which shows all of the default values which you have set which are different from the default values delivered by Innovation. This includes defaults which may have been changed at some other time. Options which have not had their default values changed will not be displayed.

RESETTING GLOBAL OPTIONS TO DELIVERED DEFAULT VALUES

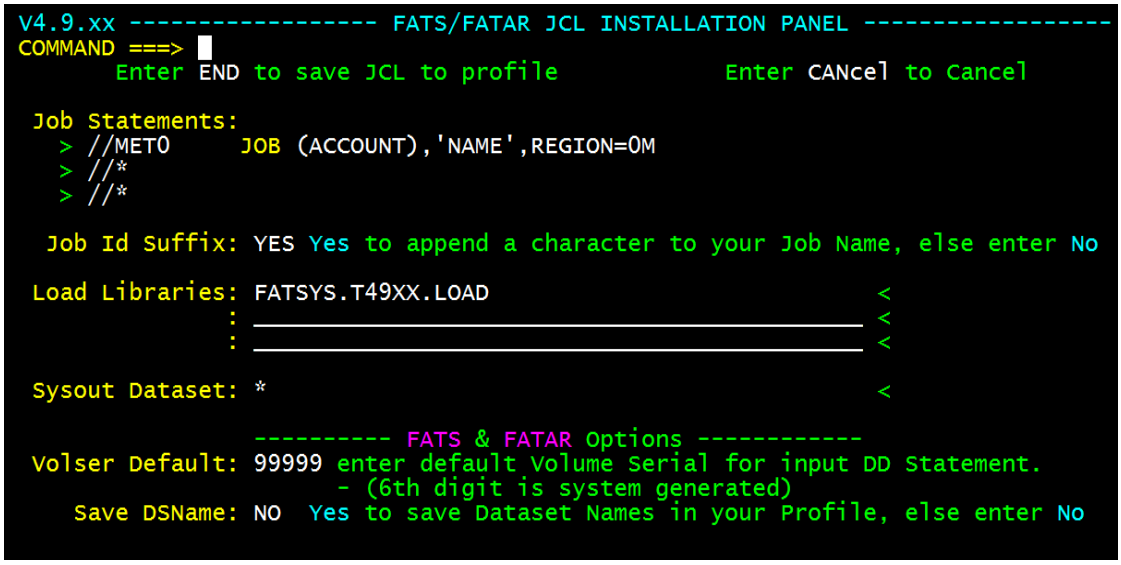

V4.9.xx FATSCOPY INSTALLATION -- SET FATSCOPY GLOBAL OPTIONS PRIMARY MENU
V4.9.xx --------- FATSCOPY / RESET GLOBAL OPTIONS CONFIRM PANEL ---------
OPTIONS==>
Please Press ENTER to confirm the RESET of all the FATSCOPY Global Options
RESET will restore all option settings back to their original
default values as from the distribution tape.
Press PF3 (END key) to cancel the RESET request.
CANCEL - Exit without saving changes
AUDIT - Display user changed options
RESET - Re-Initialize all options
FATSCOPY LIBRARY: FATSYS.FAT49XX.LOAD
<

Resetting Global Options to Delivered Default Values

If you want to reset the FATSCOPY keyword default values to their delivered default values (the values displayed with an asterisk (*) in Section 32 of this manual), enter RESET on the FATSCOPY Global Options Primary Menu. If you confirm that you want to reset the option settings, the FATSCOPY Global Options Table will be reset to its original delivered settings.

CAUTION: RESET clears all of the tape management information required by FATSCOPY. After using RESET, you must reset the TMSIN= value (Menu Option 2). If using TLMS, you must also set the VMFDSN= parameter (Menu Option 4).

JCL INSTALLATION PANEL


V4.9.xx ------------------- FATS/FATAR JCL INSTALLATION PANEL ------------------
COMMAND ===>
Enter END to save JCL to profile Enter CANcel to Cancel
Job Statements:
> //MET0 JOB (ACCOUNT),'NAME',REGION=0M
> //*
> //*
Job Id Suffix: YES Yes to append a character to your Job Name, else enter No
Load Libraries: FATSYS.T49XX.LOAD
Sysout Dataset: *
---------- FATS & FATAR Options ------------
Volser Default: 99999 enter default Volume Serial for input DD Statement.
- (6th digit is system generated)
Save DSName: NO Yes to save Dataset Names in your Profile, else enter No

Option J – JCL Installation Panel

The Installation panel (option J from the Main Menu) is used to set JCL defaults in each individual ISPF user's profile. This needs to be done by each user. Enter any JOB statement parameters required by your installation for running FATSCOPY batch jobs. If FATSCOPY is in the system link list, no Load Library is required; if not, provide the name of the FATSCOPY load library so that the correct STEPLIB DD statements can be generated. You can also change the parameter for SYSOUT= statements.

When you are done, enter END (PF3) to return to the Main Menu.

You may now select Option 3 from the Main Menu to proceed to the FATSCOPY Selection Menu, shown on the next page.

FATSCOPY SELECTION MENU

```
V4.9.xx ----------------------- FATSCOPY SELECTION MENU -----------------------
OPTION ===>

       1. Select ALL datasets by tape Volume Range

       2. Select ALL datasets by tape Volume List

       3. Select/Exclude Cataloged datasets

       R. Review Results of Simulation(s)

       A. View Audit Report

       J. Set up FATSCOPY JCL parameters

       Q. Query TMS - Tape Management Id: RMM
```

FATSCOPY Selection Menu

On this panel, Option Q can be used to display information for a volume recorded in the Tape Management System shown.

The Option J panel is used to set up default JCL parameters for your TSO USERID. This panel is automatically displayed the first time you attempt to use most of the FATSCOPY panels. The Option J panel shown below.

FATSCOPY DEFAULT JCL OPERANDS

```
V4.9.xx ------------------------ FATSCOPY JCL OPERANDS ------------------------
COMMAND ===>
FAT00CK "TAPEOUT" unit, for FATSCOPY output devices, is required

Ipt VTS Unit: ==> _______ <
Tapeout Unit: ==> _______ <

Sysout Data Sets:
     Selrpt   ==> SYSOUT=*                                                  <
     Copyrpt  ==> SYSOUT=*                                                  <
     Errorrpt ==> SYSOUT=*                                                  <
     Tapesumm ==> SYSOUT=*                                                  <
     Sysprint ==> SYSOUT=*                                                  <
     Sysabend ==> SYSOUT=*                                                  <
Space Parameter:  Cyl/Trk  Prim   Scnd   Unit
     Auditrpt ==> CYL         1      1   SYSALLDA
     Dsntable ==> CYL         1      1   SYSALLDA
     Selrpt   ==> CYL         1      1   SYSALLDA
     Echorpt  ==> CYL         1      1   SYSALLDA
     Errorrpt ==> CYL         1      1   SYSALLDA
     Sysin    ==> CYL         1      1   SYSALLDA
```

FATSCOPY Default JCL Operands

You must specify the TAPEOUT unit to be used to allocate volumes for output tape data sets. If you have an IBM virtual tape system and will be selecting input data sets by physical volume, you also need to enter the unit type for your VTS device (Ipt VTS Unit).

You can also update the Space Parameters for several data sets. The most commonly updated ones are AUDITRPT and DSNTABLE. When done, enter END (PF3) to return to the Selection Menu.

FATSCOPY ALLDSN FUNCTION ENTRY PANELS

If you select Option '1' from the FATSCOPY Selection Menu, the following panel will be displayed:

```
V4.9.xx --------------- FATSCOPY: "ALLDSN" FUNCTION ENTRY PA Row 1 to 13 of 20
COMMAND ==>
        SIM execute FATSCOPY Simulation              COPY execute FATSCOPY Copy
        END return to FATSCOPY Menu                  Q for RMM  Query
Vol. Range: from ______ incrementing by 1 for a max. of ____ not to exceed 1000
          ------------------ Selection Operands ----------------
ABRARC    - Copy ABR Archive or Application Backup files... NO    (Yes,No)
ABRINDEX  - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7     (0-999)
            ( 999 means stack all files together regardless of exp date)
EXPIRED   - Copy expired datasets (according to Tape Mgmt). YES   (Yes,No)
GROUPBY   - Group output by Volset or by Expdate ..... VOLSET
HSMML2    - Copy HSM Migration Level 2 Volumes..... NO    (Yes,No)
IMAGE     - Make an IMAGE copy of the input volume..... ___   (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000  (1-65535)
MULTIVOL  - Copy multi-volume file datasets..... YES   (Yes,No,Only)
OFFSITE   - Copy offsite (vaulted) tape volumes..... NO    (Yes,No,Only)
          ----------------------- Exclusion Operands -------------------------
VOLUMES   - Enter fully qualified Vsn's to be excluded from the Selected Range
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
                                    ______
```

Option 1 Panel: ALLDSN Volume Range

This panel allows you to begin a simulation or copy job using a range of volumes, or to query tape management for a volume. In the lower section of the panel you can specify individual volumes to be excluded from within this range. Subsequent panels will allow you to specify Exclusion Operands to bypass data sets created or last used by a specific program, and to further limit the data sets selected from each volume. Depending on the option you specify on the command line, a pop-up panel will be displayed offering you the choice of running an online simulation, submitting a batch simulation or copy job, editing the JCL for a batch job, or viewing the control statements generated.

If you enter PF3 from here, you are returned to the FATSCOPY Selection Menu.

If you select Option '2' from the FATSCOPY Selection Menu, the following panel will be displayed:

```
V4.9.xx -------------- FATSCOPY - "ALLDSN" FUNCTION ENTRY P Row 1 to 13 of 100
COMMAND ===>
        SIM execute FATSCOPY Simulation              COPY execute FATSCOPY Copy
        END return to FATSCOPY Menu                  Q for RMM  Query
          ----------------- Selection Operands ----------------
ABRARC    - Copy ABR Archive or Application Backup files... NO    (Yes,No)
ABRINDEX  - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7     (0-999)
            ( 999 means stack all files together regardless of exp date)
EXPIRED   - Copy expired datasets (according to Tape Mgmt). YES   (Yes,No)
GROUPBY   - Group output by Volset or by Expdate ........... VOLSET
HSMML2    - Copy HSM Migration Level 2 Volumes.............. NO    (Yes,No)
IMAGE     - Make an IMAGE copy of the input volume......... NO    (Yes,No)
MAXTOTFILE- Maximum number of input files to copy.......... 1000  (1-65535)
MULTIVOL  - Copy multi-volume file datasets................. YES   (Yes,No,Only)
OFFSITE   - Copy offsite (vaulted) tape volumes............ NO    (Yes,No,Only)
VIRTTYPE  - Type of VTS that contains volumes to be copied. ___   (IBM/STK)
                                                 Selection  Limits   Physical
VOLUMES   - Enter fully qualified volsers: Volume     (Y)      Have     (P/A)
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
                                           ______      _                _
```

Option 2 Panel: ALLDSN Volume List

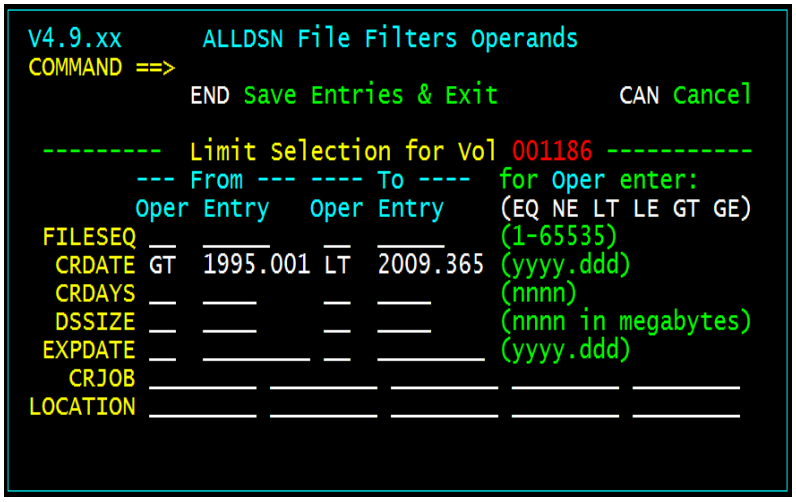
This panel allows you to begin a simulation or copy job using a list of volumes, or to query tape management for a volume. Entering a 'Y' under the Selection Limits column next to one of those volsers will allow you to limit the data sets selected from that *individual* volume.

If the volume you are specifying is a *physical* volume in a VTS, rather than a logical volume, enter a 'P' (for PHYSVOL) in the Physical column. FATSCOPY will select all the logical volumes on that physical tape.

If the volume you are specifying is a *logical* volume in a VTS, and you want FATSCOPY to determine which physical tape it is located on and select *all* the logical volumes on that physical tape, enter an 'A' (for ALLPHYS) in the Physical column.

A subsequent panel will allow you to specify additional Exclusion Operands that will apply to *all* of the volumes, to bypass data sets created or last used by a specific program.

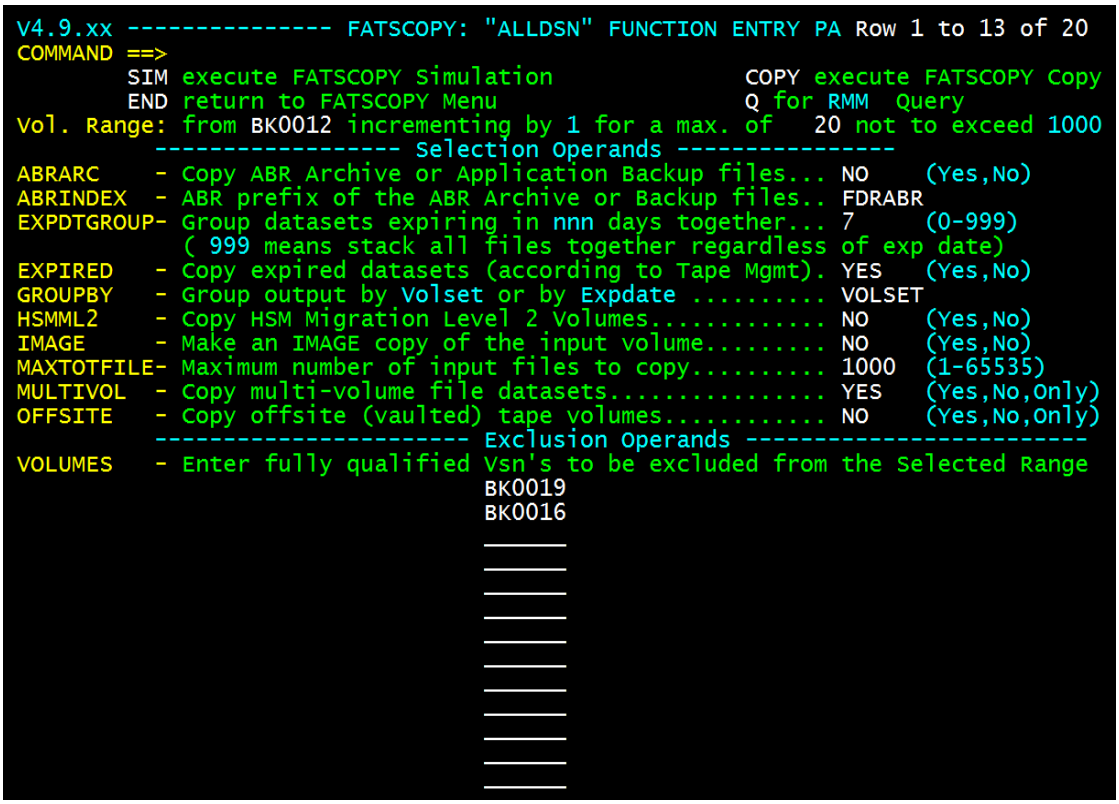
For example, if you enter the volser 001186 and then 'Y' under Selection Limits, followed by the Enter key, a pop-up panel will be displayed, where you can enter criteria to limit the selection, such as a range of data set creation dates:


V4.9.xx ALLDSN File Filters Operands
COMMAND ==>
END Save Entries & Exit CAN Cancel
--------- Limit Selection for Vol 001186 -----------
--- From --- ---- To ---- for Oper enter:
Oper Entry Oper Entry (EQ NE LT LE GT GE)
FILESEQ (1-65535)
CRDATE GT 1995.001 LT 2009.365 (yyyy.ddd)
CRDAYS (nnnn)
DSSIZE (nnnn in megabytes)
EXPDATE (yyyy.ddd)
CRJOB
LOCATION

In this example, FATSCOPY is instructed to select from volume 001186 only the data sets that were created between January 2, 1995 and December 30, 2009. When you hit the END key (usually PF3), you will be returned to the previous panel where you can add more volumes to be copied or submit a job for execution.

ALLDSN EXAMPLES

This example uses option 1 from the FATSCOPY Selection Menu.


V4.9.xx --------------- FATSCOPY: "ALLDSN" FUNCTION ENTRY PA Row 1 to 13 of 20
COMMAND ==>
SIM execute FATSCOPY Simulation
COPY execute FATSCOPY Copy
END return to FATSCOPY Menu
Q for RMM Query
Vol. Range: from BK0012 incrementing by 1 for a max. of 20 not to exceed 1000
Selection Operands
ABRARC - Copy ABR Archive or Application Backup files... NO (Yes,No)
ABRINDEX - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7 (0-999)
(999 means stack all files together regardless of exp date)
EXPIRED - Copy expired datasets (according to Tape Mgmt). YES (Yes,No)
GROUPBY - Group output by Volset or by Expdate ..... VOLSET
HSMML2 - Copy HSM Migration Level 2 Volumes..... NO (Yes,No)
IMAGE - Make an IMAGE copy of the input volume..... NO (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000 (1-65535)
MULTIVOL - Copy multi-volume file datasets..... YES (Yes,No,Only)
OFFSITE - Copy offsite (vaulted) tape volumes..... NO (Yes,No,Only)
Exclusion Operands
VOLUMES - Enter fully qualified Vsn's to be excluded from the Selected Range
BK0019
BK0016

Volume Range Example

In the example above, a range of 20 volumes beginning with volume BK0012 will be selected (when SIM is specified on the command line) or copied (when COPY is specified). However, volumes BK0019 and BK0016 will not be used, because they were excluded from the specified range. Note that if any of the selected volumes are part of multivolume sets, the other volumes which are part of those multivolume sets will also be selected (even if they are not within the specified volume range).

If these were *physical* volsers in a virtual tape system, rather than logical volumes, you would enter “Phys” for PHYSVOL and enter the VTS device type on the VIRTTYPE entry line.

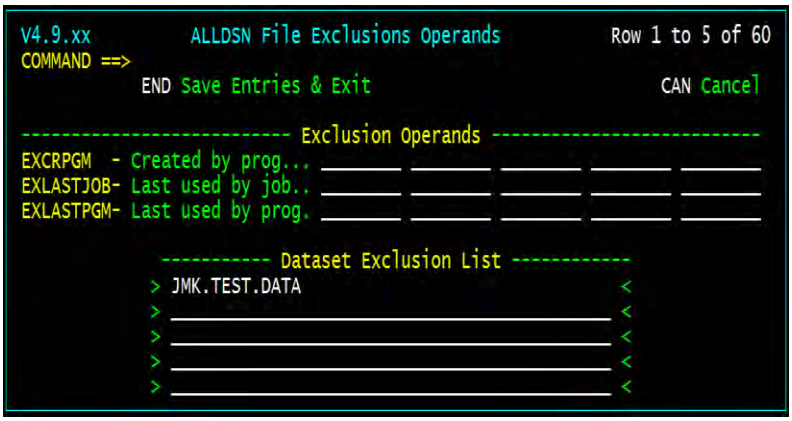
If these are *logical* volsers in a VTS, and you want FATSCOPY to determine which physical tapes they are located on and select *all* the logical volumes on those physical tapes, enter “Log” for PHYSVOL and enter the VTS device type on the VIRTTYPE entry line.

This example uses option 2 from the FATSCOPY Selection Menu.

V4.9.xx ---------- FATSCOPY - "ALLDSN" FUNCTION ENTRY P Row 1 to 13 of 100
COMMAND ===>
SIM execute FATSCOPY Simulation
COPY execute FATSCOPY Copy
END return to FATSCOPY Menu
Q for RMM Query
Selection Operands
ABRARC - Copy ABR Archive or Application Backup files... NO (Yes,No)
ABRINDEX - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7 (0-999)
(999 means stack all files together regardless of exp date)
EXPIRED - Copy expired datasets (according to Tape Mgmt). YES (Yes,No)
GROUPBY - Group output by Volset or by Expdate ..... VOLSET
HSMML2 - Copy HSM Migration Level 2 Volumes..... NO (Yes,No)
IMAGE - Make an IMAGE copy of the input volume..... NO (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000 (1-65535)
MULTIVOL - Copy multi-volume file datasets..... YES (Yes,No,Only)
OFFSITE - Copy offsite (vaulted) tape volumes..... NO (Yes,No,Only)
VIRTTYPE - Type of VTS that contains volumes to be copied. ___ (IBM/STK)
Selection Limits Physical
VOLUMES - Enter fully qualified volsers: Volume (Y) Have (P/A)
111076
ABC001
ABC002
001186

Volume List Example

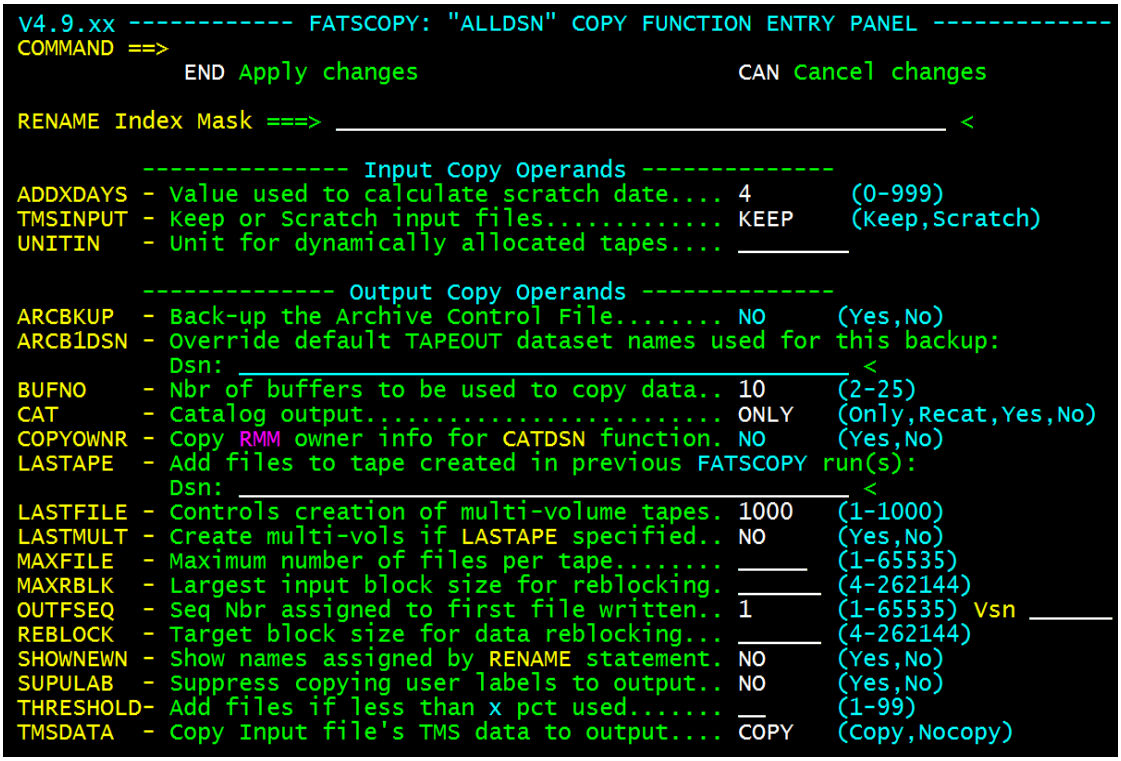
In the example above, 4 volumes will be selected (when SIM is specified on the command line) or copied (when COPY is specified). If COPY is specified, the following pop-up panel will be displayed:


V4.9.xx ALLDSN File Exclusions Operands Row 1 to 5 of 60
COMMAND ==>
END Save Entries & Exit CAN Cancel
Exclusion Operands
EXCRPGM - Created by prog...
EXLASTJOB- Last used by job..
EXLASTPGM- Last used by prog.
Dataset Exclusion List
> JMK.TEST.DATA <

ALLDSN Exclusion Panel

Enter the data sets that you want to be excluded from being selected; then hit the ENTER key to accept the entries. In this case, all data sets on volumes 111076, ABC001, ABC002, and 001186 will be selected, except for any named JMK.TEST.DATA.

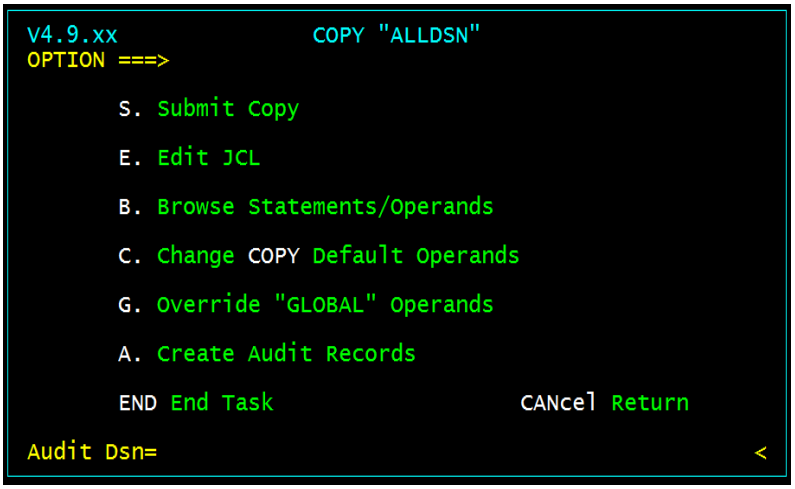
Hit the END key (usually PF3), and the following panel will be displayed:


V4.9.xx FATSCOPY: "ALLDSN" COPY FUNCTION ENTRY PANEL
COMMAND ==>
END Apply changes CAN Cancel changes
RENAME Index Mask ===>
Input Copy Operands
ADDXDAYS - Value used to calculate scratch date.... 4 (0-999)
TMSINPUT - Keep or Scratch input files..... KEEP (Keep,Scratch)
UNITIN - Unit for dynamically allocated tapes....
Output Copy Operands
ARCBKUP - Back-up the Archive Control File..... NO (Yes,No)
ARCB1DSN - Override default TAPEOUT dataset names used for this backup:
Dsn:
BUFNO - Nbr of buffers to be used to copy data.. 10 (2-25)
CAT - Catalog output..... ONLY (Only,Recat,Yes,No)
COPYOWNR - Copy RMM owner info for CATDSN function. NO (Yes,No)
LASTAPE - Add files to tape created in previous FATSCOPY run(s):
Dsn:
LASTFILE - Controls creation of multi-volume tapes. 1000 (1-1000)
LASTMULT - Create multi-vols if LASTAPE specified.. NO (Yes,No)
MAXFILE - Maximum number of files per tape..... (1-65535)
MAXRBLK - Largest input block size for reblocking. (4-262144)
OUTFSEQ - Seq Nbr assigned to first file written.. 1 (1-65535) Vsn
REBLOCK - Target block size for data reblocking... (4-262144)
SHOWNEWN - Show names assigned by RENAME statement. NO (Yes,No)
SUPULAB - Suppress copying user labels to output.. NO (Yes,No)
THRESHOLD- Add files if less than x pct used..... (1-99)
TMSDATA - Copy Input file's TMS data to output.... COPY (Copy,Nocopy)

ALLDSN Copy Function Entry Panel

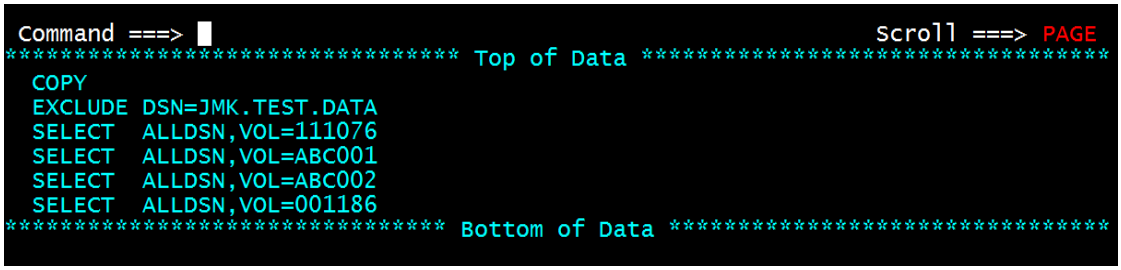
You can change common FATSCOPY control statement parameters from their default values (for this job only) using this panel.

After making any changes you want, hit END to proceed to the following pop-up panel:



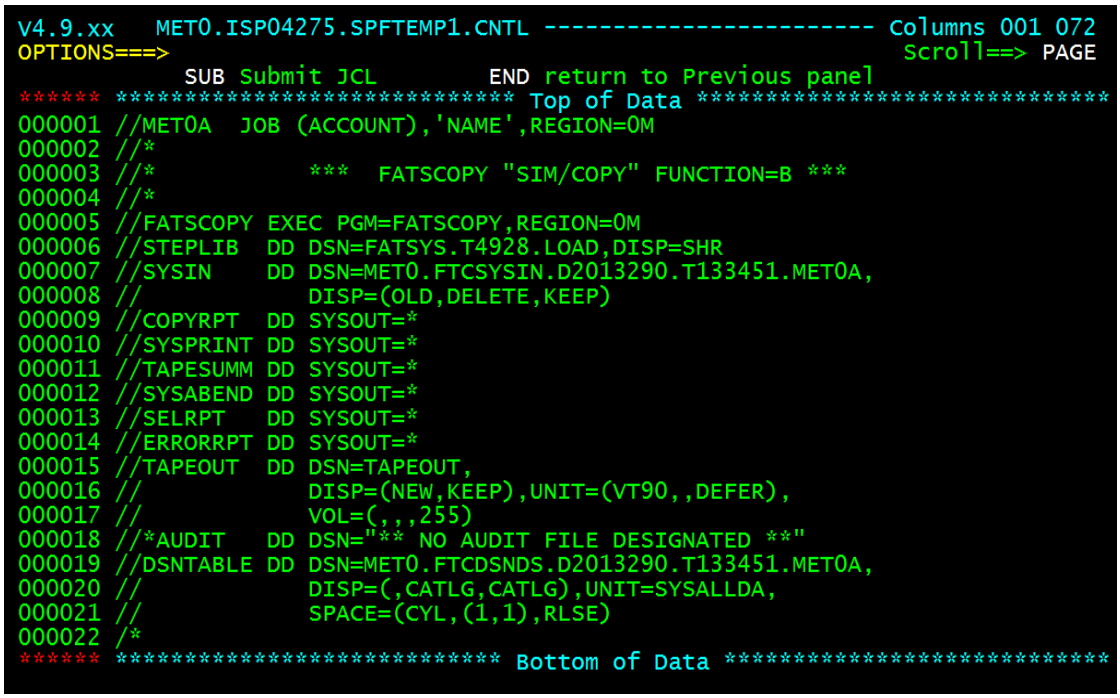
ALLDSN Copy Option Panel

Some of the options displayed in this pop-up menu are explained below in the next several displays.



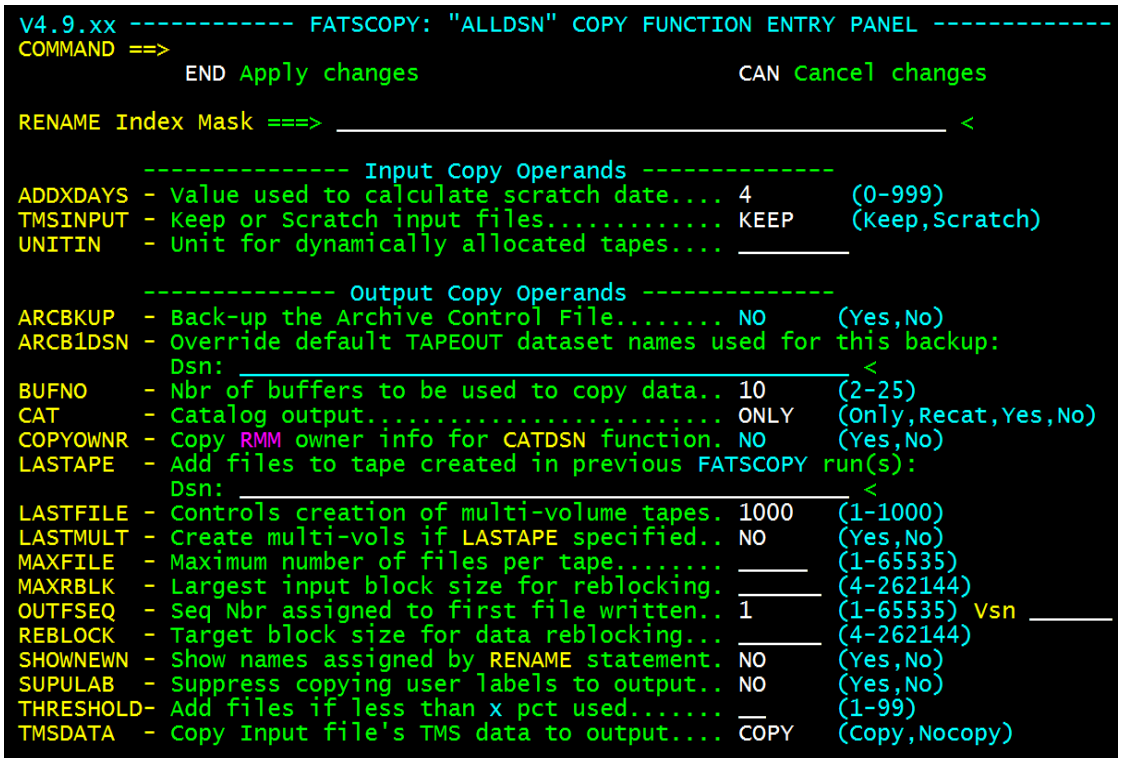
Option B: Browse

Selecting Option B from the pop-up menu will show the control statements FATSCOPY will use when the copy job is submitted. Entering PF3 returns you to the pop-up menu.


V4.9.xx MET0.ISP04275.SPFTEMP1.CNTL ---------- Columns 001 072
OPTIONS===> Scroll==> PAGE
SUB Submit JCL END return to Previous panel
****** ********** Top of Data **********
000001 //MET0A JOB (ACCOUNT),'NAME',REGION=0M
000002 //*
000003 //* *** FATSCOPY "SIM/COPY" FUNCTION=B ***
000004 //*
000005 //FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
000006 //STEPLIB DD DSN=FATSYS.T4928.LOAD,DISP=SHR
000007 //SYSIN DD DSN=MET0.FTCSYSIN.D2013290.T133451.MET0A,
000008 // DISP=(OLD,DELETE,KEEP)
000009 //COPYRPT DD SYSOUT=*
000010 //SYSPRINT DD SYSOUT=*
000011 //TAPESUMM DD SYSOUT=*
000012 //SYSABEND DD SYSOUT=*
000013 //SELRPT DD SYSOUT=*
000014 //ERRORRPT DD SYSOUT=*
000015 //TAPEOUT DD DSN=TAPEOUT,
000016 // DISP=(NEW,KEEP),UNIT=(VT90,,DEFER),
000017 // VOL=(,,,255)
000018 //*AUDIT DD DSN="** NO AUDIT FILE DESIGNATED **"
000019 //DSNTABLE DD DSN=MET0.FTCDSNDS.D2013290.T133451.MET0A,
000020 // DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
000021 // SPACE=(CYL,(1,1),RLSE)
000022 /*
****** ********** Bottom of Data **********

Option E: Edit

The above panel will be displayed if you select Option E from the pop-up panel. You can modify the JCL, although you should not change the DSN on the SYSIN DD. If you wish to run the job, you can enter SUB. To return to the previous panel, enter END and hit enter or press the END key (usually PF3 or PF15).


V4.9.xx ------------ FATSCOPY: "ALLDSN" COPY FUNCTION ENTRY PANEL -------------
COMMAND ==>
END Apply changes CAN Cancel changes
RENAME Index Mask ===>
---------- Input Copy Operands ----------
ADDXDAYS - Value used to calculate scratch date.... 4 (0-999)
TMSINPUT - Keep or Scratch input files..... KEEP (Keep,Scratch)
UNITIN - Unit for dynamically allocated tapes....
---------- Output Copy Operands ----------
ARCBKUP - Back-up the Archive Control File..... NO (Yes,No)
ARCB1DSN - Override default TAPEOUT dataset names used for this backup:
Dsn:
BUFNO - Nbr of buffers to be used to copy data.. 10 (2-25)
CAT - Catalog output..... ONLY (Only,Recat,Yes,No)
COPYOWNR - Copy RMM owner info for CATDSN function. NO (Yes,No)
LASTAPE - Add files to tape created in previous FATSCOPY run(s):
Dsn:
LASTFILE - Controls creation of multi-volume tapes. 1000 (1-1000)
LASTMULT - Create multi-vols if LASTAPE specified.. NO (Yes,No)
MAXFILE - Maximum number of files per tape..... (1-65535)
MAXRBLK - Largest input block size for reblocking. (4-262144)
OUTFSEQ - Seq Nbr assigned to first file written.. 1 (1-65535) Vsn
REBLOCK - Target block size for data reblocking... (4-262144)
SHOWNEWN - Show names assigned by RENAME statement. NO (Yes,No)
SUPULAB - Suppress copying user labels to output.. NO (Yes,No)
THRESHOLD- Add files if less than x pct used..... (1-99)
TMSDATA - Copy Input file's TMS data to output.... COPY (Copy,Nocopy)

Option C: Change COPY Default Operands

Selecting Option C from the pop-up panel will display the Copy Function Entry Panel that you saw earlier. You can again change most FATSCOPY control statement parameters using the options on this panel. These changes are in effect for this job only.

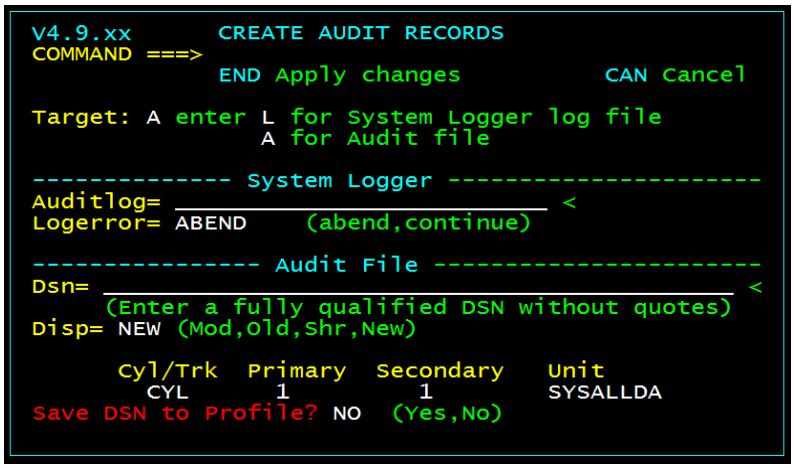
If you enter END or press the END key, the FATSCOPY control statements will be modified and you will be returned to the Copy options menu. You can verify the changes by selecting Option B from the pop-up-menu.

```
V4.9.xx ---------------------- FATSCOPY "GLOBAL" OPERANDS ----------------------
COMMAND ===>
             END Save Entries & Exit                                  CAN Cancel


DUMP       - Produce an ABEND dump for unusual abends.......... NO  (Yes,No,Ask)
ECHO       - Echo the FATSCOPY option values in SYSPRINT............ NO  (Yes,No)
HSMBAKMASK- Mask to exclude backup tape datasets created by DFSMShsm:
                > _____________________________________________ <
HSMMIGMASK- Mask to exclude migration tape datasets created by DFSMShsm:
                > _____________________________________________ <
LINECNT - Number of lines per page on FATSCOPY reports......... 60    (1,32767)
MAJERR  - stop processing after 1     major errors are encountered    (1,32767)
MAXERR  - Stop processing after 1     permanent data checks detected  (1,32767)
MAXTERR - Stop processing after 100   temporary data checks detected  (1,32767)
NOTCATRC- Retcode when cataloging bypassed for uncataloged datasets.. 0 (0,4,8)
OPCAP   - Uncompressed Gigabyte capacity of output tapes......... 10   (1-1000)
RETRY     Number of RMM update attempts before terminating......... 0   (0-255)
RMMFAIL - How to proceed when RMM update fails.. CONTINUE (Continue,Stop,Retry)
SELTERR - Set Cond Code if SELECT/EXCLUDE stmts never referenced.. YES (Yes,No)
ABEND   - Abnormal termination Option .................. ABEND   (ABEND,Retcode)
```

Option G: Global Operands

Selecting Option G from the pop-up panel will display the Global Operands Panel. This panel allows you to change some less-commonly modified FATSCOPY control statement parameters. These changes are in effect for this job only.

If you enter END or press the END key, the FATSCOPY control statements will be modified and you will be returned to the Copy options menu. You can verify the changes by selecting Option B from the pop-up-menu.



Option A: Create Audit Records

Selecting Option A from the pop-up panel will display the Create Audit Records panel. FATSCOPY can write audit records to either a sequential data set (specify “A” for Target, and enter Dsn= and Disp= values), or to a system logger data set (specify “L” for Target, and enter the name of an active log stream file on the Auditlog= line).

If you enter YES for “Save DSN to Profile?”, these values will be saved in the ISPF profile for your userid. Enter END to continue using the values you have entered and return to the Copy options menu.

At this point, you can enter “S” if you are ready to submit the Copy job.

A SAMPLE ALLDSN JOB

```
V4.9.xx -------------- FATSCOPY - "ALLDSN" FUNCTION ENTRY P Row 1 to 13 of 100
COMMAND ===> COPY
        SIM execute FATSCOPY Simulation              COPY execute FATSCOPY Copy
        END return to FATSCOPY Menu                  Q for RMM  Query
          ------------------ Selection Operands -----------------
ABRARC     - Copy ABR Archive or Application Backup files... NO    (Yes,No)
ABRINDEX   - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 999   (0-999)
            ( 999 means stack all files together regardless of exp date)
EXPIRED    - Copy expired datasets (according to Tape Mgmt). YES   (Yes,No)
GROUPBY    - Group output by Volset or by Expdate ........... VOLSET
HSMML2     - Copy HSM Migration Level 2 Volumes.............. NO    (Yes,No)
IMAGE      - Make an IMAGE copy of the input volume.......... NO    (Yes,No)
MAXTOTFILE- Maximum number of input files to copy.......... 1000  (1-65535)
MULTIVOL   - Copy multi-volume file datasets................. YES   (Yes,No,Only)
OFFSITE    - Copy offsite (vaulted) tape volumes............. NO    (Yes,No,Only)
VIRTTYPE   - Type of VTS that contains volumes to be copied. ___   (IBM/STK)
                                                  Selection  Limits   Physical
VOLUMES    - Enter fully qualified volsers: Volume    (Y)      Have     (P/A)
                                            003392      _                _
                                            003475      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
                                            ______      _                _
```

ALLDSN Function Entry Panel

If you specify COPY on command line in this panel and hit Enter, a pop-up menu for Exclusion Operands will appear. Hitting End brings you to the “ALLDSN” Copy Function Entry Panel, where you can change operand defaults. Hit End again to bring up the Copy “ALLDSN” pop-up menu; Option S from that menu will submit a batch job to copy the data sets selected from volumes 003392 and 003475. Copies of the reports generated by the batch job follow.

ALLDSN SAMPLE OUTPUT

```
FATSCOPY  INNOVATION TAPE COPY UTILITY   VER 4.9.XX       -- INNOVATION DATA PROCESSING  AUTHORIZED  10/04/2011 PAGE   1

FATS070 CONTROL CARD TABLE SIZE IS  32768 BYTES

FATSCOPY CONTROL CARDS

   1--COPY  EXPDTGROUP=999
   2--  SELECT  ALLDSN,VOL=003392
   3--  SELECT  ALLDSN,VOL=003475

FATS071 TAPE BUFFER SIZE IS 262144 BYTES.  NUMBER OF BUFFERS IS 10

FATS072 TAPEIN DATA WILL BE COPIED TO TAPEOUT

                                          FATSCOPY FILE TOTALS
 DATA SET NAME                          FILE BLOCKS     FILE BYTES     OUTPUT BLOCKS     OUTPUT BYTES
 MET.FATCOPY.ASMV50                           12547       60727480             12547         60727480

FATS132 **TAPE MGMT INFO FOR DSN=MET.FATCOPY.ASMV50                        ON VOL=CCR216 WAS COPIED FROM VOL=003392

FATS133 **EXPIRATION DATE FOR DSN=MET.FATCOPY.ASMV50                       ,VOL=CCR216 CHANGED TO  1999365

                                          FATSCOPY FILE TOTALS
 DATA SET NAME                          FILE BLOCKS     FILE BYTES     OUTPUT BLOCKS     OUTPUT BYTES
 MET.FATCOPY.CBT1085                           5648      202641507              5648        202641507

FATS132 **TAPE MGMT INFO FOR DSN=MET.FATCOPY.CBT1085                       ON VOL=CCR216 WAS COPIED FROM VOL=003475

FATS133 **EXPIRATION DATE FOR DSN=MET.FATCOPY.CBT1085                      ,VOL=CCR216 CHANGED TO  1999365

FATS165 **LAST USED INFO FOR VOLUME=CCR216 COPIED FROM VOLUME=003475

FATS166 **EXPIRATION DATE FOR OUTPUT VOLUME=CCR216 HAS BEEN SET TO: 1999365

FATS020** COPY FUNCTION COMPLETE. ALL VOLUMES PROCESSED

                                              FINAL TOTALS
    FILES READ  BLOCKS READ       BYTES READ  TEMP ERRS  PERM ERRS  FILES WRITTEN  BLOCKS WRITTEN      BYTES WRITTEN

             2        18195        263368987          0          0              2           18195          263368987
```

FATSCOPY SYSPRINT Report

For each data set copied, this report displays the data set name, the number of blocks and files in the data set, the number of output blocks (which should always be the same as the input blocks, unless REBLOCK was specified), whether or not the output tape was cataloged, whether the tape management information was propagated from the input to the output, and the expiration date that was assigned to the output. The Final Totals report shows file, blocks, and bytes read, the number of data check errors that occurred on the input, and the number of files and blocks written to the output.

```
                                FATSCOPY DATA SET SELECTION REPORTS                    10/04/2011 PAGE      1

DATA SET NAME                            VOLSER FSEQ  VSQ DEVT   CRDATE  EXDATE  CREATJOB TMSSTAT SIZE   STATUS REASON   PHYVOL
---------------------------------------- ------ ----- --- ------ ------- ------- -------- ------- ------ ------ -------- ------
MET.FATCOPY.ASMV50                       003392     1   1 3590-1 2001093 1999365 FATSQA2  MASTER     60M SELECT
MET.FATCOPY.CBT1085                      003475     1   1 3590-1 2001051 2012366 METFATC9 MASTER    185M SELECT
```

FATSCOPY Data Set Selection Report

This report shows information for all of the data sets selected for copying or that were bypassed because they failed to meet all of the selection criteria specified. The information in the report comes directly from tape management.

```
                              FATSCOPY DATA SET COPY/SIMULATION REPORT                10/04/2011 PAGE      1

DATA SET NAME                            VOLSER FSEQ  VSEQ  EXDATE  GROUP FUNC O/P EXP O/P VOL I/P  TMINFO OSEQ  RC
---------------------------------------- ------ ----- ----  ------- ----- ---- ------- ------- ---- ------ ----- ----
MET.FATCOPY.ASMV50                       003392     1    1  1999365     1 COPY 1999365 CCR216  KEEP COPY       1 0000
MET.FATCOPY.CBT1085                      003475     1    1  2012366     1 COPY 2012366 CCR216  KEEP COPY       2 0000
```

FATSCOPY Data Set Copy/Simulation Report

The report above shows information for all the data sets that were copied. Data sets with the same number under the GROUP heading will be stacked together. In the example above the KEEP under the I/P heading shows that the input data sets was retained because TMSINPUT=KEEP was specified or taken as a default. If TMSINPUT=SCRATCH was specified and FATSCOPY was able to change the expiration date for the input, SCR would appear under the I/P heading. The TMINFO column shows whether the tape management information was propagated from the input to the output. The RC column shows the highest return code FATSCOPY issued when copying the data set.

```
                                   FATSCOPY ERROR/MESSAGES REPORT                    10/04/2011 PAGE    1
-------------------------------------------------------------------------------------------------------------

FATS163 **CATALOGING BYPASSED FOR UNCATALOGED INPUT DATA SET. DSN=MET.FATCOPY.ASMV50                    VOL=003392
FATS163 **CATALOGING BYPASSED FOR UNCATALOGED INPUT DATA SET. DSN=MET.FATCOPY.CBT1085                   VOL=003475
```

FATSCOPY Error/Messages Report

The contents of the ERRORRPT DD are displayed in this report. FATSCOPY writes any error messages to this report if it is present in your JCL. If no ERRORRPT DD is present, the error messages will be written to SYSPRINT. Innovation recommends that you include an ERRORRPT in all of your FATSCOPY jobs. In the example above, FATSCOPY issued a FATS163 message to indicate that it was unable to catalog the output data set. In this case, the input was not cataloged, and CAT=ONLY was taken as a default so there is no error.

FATSCOPY SIMULATION

```
V4.9.xx -------------- FATSCOPY - "ALLDSN" FUNCTION ENTRY P Row 1 to 13 of 100
COMMAND ===> SIM
        SIM execute FATSCOPY Simulation              COPY execute FATSCOPY Copy
        END return to FATSCOPY Menu                  Q for RMM  Query
          ------------------ Selection Operands ----------------
ABRARC    - Copy ABR Archive or Application Backup files... NO    (Yes,No)
ABRINDEX  - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7     (0-999)
            ( 999 means stack all files together regardless of exp date)
EXPIRED   - Copy expired datasets (according to Tape Mgmt). YES   (Yes,No)
GROUPBY   - Group output by Volset or by Expdate ........... VOLSET
HSMML2    - Copy HSM Migration Level 2 Volumes.............. NO    (Yes,No)
IMAGE     - Make an IMAGE copy of the input volume......... NO    (Yes,No)
MAXTOTFILE- Maximum number of input files to copy.......... 1000  (1-65535)
MULTIVOL  - Copy multi-volume file datasets................ YES   (Yes,No,Only)
OFFSITE   - Copy offsite (vaulted) tape volumes............ NO    (Yes,No,Only)
VIRTTYPE  - Type of VTS that contains volumes to be copied. ___   (IBM/STK)
                                                  Selection  Limits   Physical
VOLUMES   - Enter fully qualified volsers: Volume    (Y)      Have     (P/A)
                                           001186     _                 _
                                           001120     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
                                           ______     _                 _
```

ALLDSN SIM Example

A FATSCOPY Simulate will be run selecting all of the data sets on volumes 001186 and 001120. If either of these volumes are part of a multi-volume set, data sets on those volumes will also be selected. Pressing ENTER then END will display the following pop-up menu:

```
V4.9.xx -------------- FATSCOPY - "ALLDSN" FUNCTION ENTRY PANEL ---------------
COMMAND ===> SIM
        SIM execute FATSCOPY Simulation              COPY execute FATSCOPY Copy
        END return to FATSCOPY Menu                  Q for RMM  Query
          ------------------ Selection Operands ----------------
ABRARC    - Copy ABR Archive or Application Backup files... NO    (Yes,No)
ABRINDEX  - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7     (0-999)
            ( 999 means stack all files together regardless of exp date)
EXPIRED   - Copy                                          gmt). YES   (Yes,No)
GROUPBY   - Group   V4.9.xx   SIMULATE "ALLDSN"          ..... VOLSET
HSMML2    - Copy    OPTION ===>                         ..... NO    (Yes,No)
IMAGE     - Make                                        ..... NO    (Yes,No)
MAXTOTFILE- Maxim    F. Foreground Simulation           ..... 1000  (1-65535)
MULTIVOL  - Copy                                        ..... YES   (Yes,No,Only)
OFFSITE   - Copy     S. Submit in Batch mode            ..... NO    (Yes,No,Only)
VIRTTYPE  - Type                                        pied. ___   (IBM/STK)
                     E. Edit JCL for Batch mode         ection  Limits   Physical
VOLUMES   - Enter                                       (Y)       Have     (P/A)
                     B. Browse Statements/Operands       _                 _
                                                         _                 _
                     G. Override "GLOBAL" Operands       _                 _
                                                         _                 _
                    END End Task     CANcel Return      _                 _
                                                         _                 _
                                                         _                 _
                                                ______   _                 _
                                                ______   _                 _
                                                ______   _                 _
                                                ______   _                 _
                                                ______   _                 _
                                                ______   _                 _
```

FATSCOPY ONLINE SIMULATION

If you select Option F, Foreground Simulation from the Simulate Options Pop-up Menu, FATSCOPY will run an online simulation. Note that your terminal will be locked until the simulation completes. If you specify a large number of volumes, you may want to select Option S to submit a batch job. When the online simulation completes, the Simulation Review Menu panel will be displayed.

```
V4.9.xx ------------------- "ALLDSN" SIMULATION REVIEW MENU --------------------
OPTION ===>
            END return to FATSCOPY menu


            R. VIEW SELECTION REPORT

            D. VIEW SELECTED DATASETS              ( Total Dsns=         12 )

            V. VIEW SELECTED VOLUMES               ( Total In  =          3 )
                                                   ( Est. Out  =          2 )
                                                   ( Compressed=          2 )

            M. VIEW SIMULATION MESSAGES/ERRORS

            S. VIEW SYSPRINT DATA SET


            P. PROCEED TO COPY FUNCTION
```

Simulation Review Menu

This panel shows the total number of data sets and volumes selected, the estimated number of output volumes (both compressed and uncompressed) needed to hold the output data sets, and whether or not the Foreground Simulation wrote any messages. Option O will appear only if you used the ECHO=YES option on the FATSCOPY "Global" Operands panel.

Enter Option D to view the data sets selected.

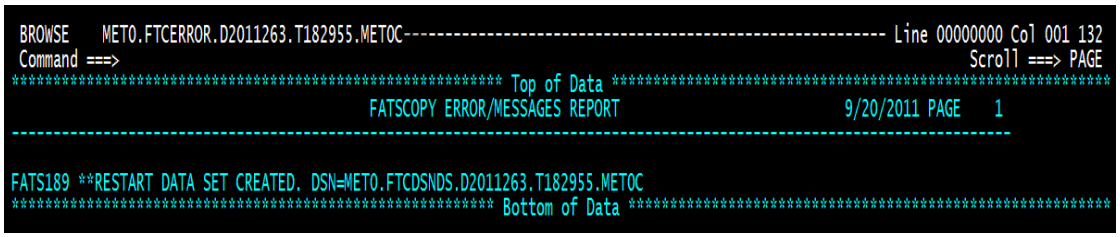
```
V4.9.xx -------- FATSCOPY - "ALLDSN" SIMULATION REPORT ----- Row 1 to 12 of 12
COMMAND ===>

Data Set Name                                FSeq ExpDate  VolSer  VSeq  Group
JMK.LABELED.FATS.FILE1                          1 2015106  001186     1      1
JMK.LABELED.FATS.FILE2                          2 2015106  001186     1      1
JMK.LABELED.FATS.FILE3                          3 2015106  001186     1      1
JMK.LABELED.FATS.FILE4                          4 2015106  001186     1      1
JMK.LABELED.FATS.FILE5                          5 2015106  001186     1      1
JMK.LABELED.FATS.FILE6                          6 2011106  001186     1      1
JMK.LABELED.FATS.FILE7                          7 2015106  001186     1      1
JMK.LABELED.FATS.FILE8                          8 2015106  001186     1      1
JMK.LABELED.FATS.FILE9                          9 2011106  001186     1      1
JMK.LABELED.FATS.FILE10                        10 2015106  001186     1      1
JMK.CPK.TEST.BACKUP                             1 2020365  001120     1      2
JMK.CPK.TEST.BACKUP                             1 2020365  001121     2      2
******************************** Bottom of data ********************************
```

Option D: Data Set View

The simulation report will show the data sets selected, the volume that contains each data set, the file sequence number, and the expiration date of the data set. Enter END or hit the END key (usually PF3 or PF15) to return to the Simulation Review Menu.

Option V: Volume View

Entering Option V from the Simulation Review Menu panel will display a list of the volumes selected for copying and the number of data sets on each volume. Enter END or hit the END key (usually PF3 or PF15) to return to the Simulation Review Menu.



Option M: Messages View

Entering Option M from the FATSCOPY Simulation Review Menu displays the FATSCOPY Error/Messages Report. The report contains any messages issued by FATSCOPY during the online execution. The FATS189 message shows the name of the output data set from the online simulation that will be used in the batch job that copies the data sets.

```
BROWSE    MET0.FTCSELDS.D2011263.T182955.MET0A---------------------------------------------------- Line 00000000 Col 001 132
Command ===>                                                                                  Scroll ===> PAGE
******************************************************* Top of Data ********************************************************
                                  FATSCOPY DATA SET SELECTION REPORTS                      9/20/2011 PAGE      1

DATA SET NAME                          VOLSER FSEQ  VSQ DEVT   CRDATE  EXDATE  CREATJOB TMSSTAT SIZE   STATUS REASON   PHYVOL
-------------------------------------- ------ ----- --- ------ ------- ------- -------- ------- ------ ------ -------- ------
JMK.LABELED.FATS.FILE1                 001186     1   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE2                 001186     2   1        1999365 2012355 JMKLALB1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE3                 001186     3   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE4                 001186     4   1        1999276 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE5                 001186     5   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE6                 001186     6   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE7                 001186     7   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE8                 001186     8   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE9                 001186     9   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.LABELED.FATS.FILE10                001186    10   1        1999365 2012355 JMKLABL1 MASTER   <1.0M SELECT
JMK.CPK.TEST.BACKUP                    001120     1   1        1997318 2020365          MASTER   <1.0M SELECT
JMK.CPK.TEST.BACKUP                    001121     1   2        1997318 2020365          MASTER   <1.0M SELECT
******************************************************* Bottom of Data *****************************************************
```

Option R: Data Set Selection Report

Entering Option R from the FATSCOPY Simulation Review Menu displays the FATSCOPY Selection Report. The report shows all the data sets FATSCOPY considered during the selection process. There will be a line for each data set showing whether it was selected for copying or bypassed. If the data set was bypassed, the REASON column will contain information showing why the data set was bypassed. If you specified selection by VTS physical volume and the data set is located on one of those volumes, the physical volume it is located on is also displayed.

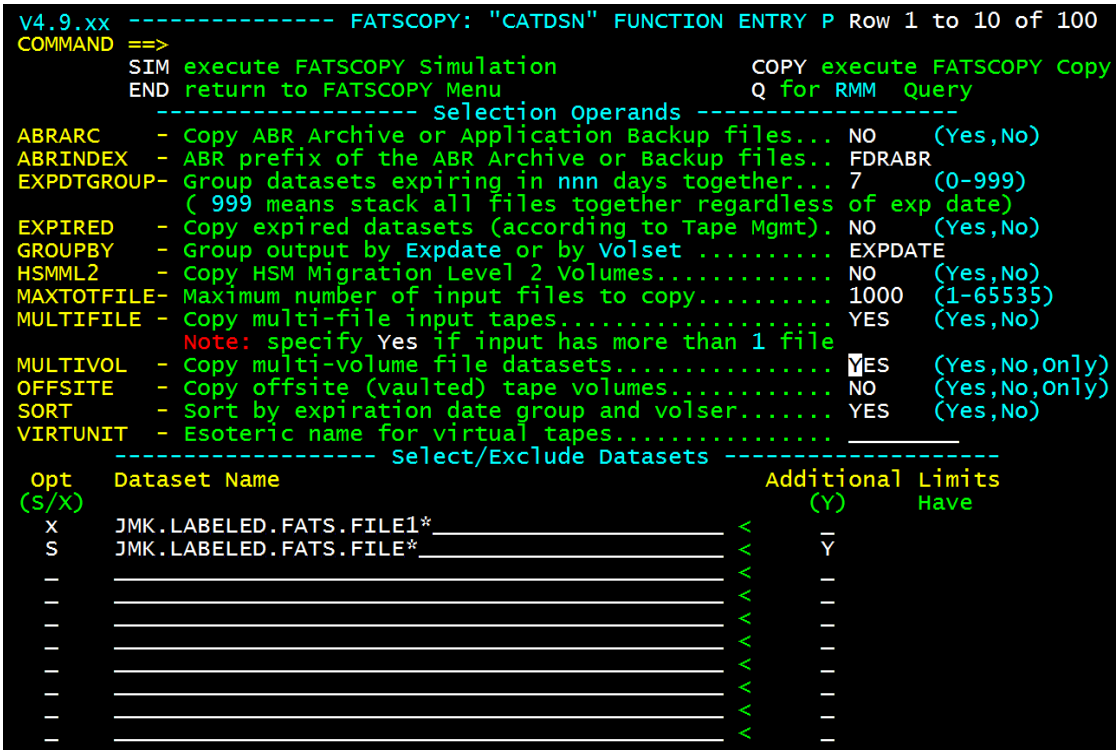
The remaining options on the Simulate ALLDSN Menu (Options S, E, B, and G) are demonstrated beginning on page 34-14. Refer to those pages for detailed description of these options.

CATDSN FUNCTION ENTRY PANEL

V4.9.xx --------------- FATSCOPY: "CATDSN" FUNCTION ENTRY P Row 1 to 10 of 100
COMMAND ==>
SIM execute FATSCOPY Simulation
COPY execute FATSCOPY Copy
END return to FATSCOPY Menu
Q for RMM Query
Selection Operands
ABRARC - Copy ABR Archive or Application Backup files... NO (Yes,No)
ABRINDEX - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7 (0-999)
(999 means stack all files together regardless of exp date)
EXPIRED - Copy expired datasets (according to Tape Mgmt). NO (Yes,No)
GROUPBY - Group output by Expdate or by Volset ..... EXPDATE
HSMML2 - Copy HSM Migration Level 2 Volumes..... NO (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000 (1-65535)
MULTIFILE - Copy multi-file input tapes..... NO (Yes,No)
Note: specify Yes if input has more than 1 file
MULTIVOL - Copy multi-volume file datasets..... YES (Yes,No,Only)
OFFSITE - Copy offsite (vaulted) tape volumes..... NO (Yes,No,Only)
SORT - Sort by expiration date group and volser..... YES (Yes,No)
VIRTUNIT - Esoteric name for virtual tapes.....
Select/Exclude Datasets
Opt (S/X)
Dataset Name
Additional Limits (Y)
Have

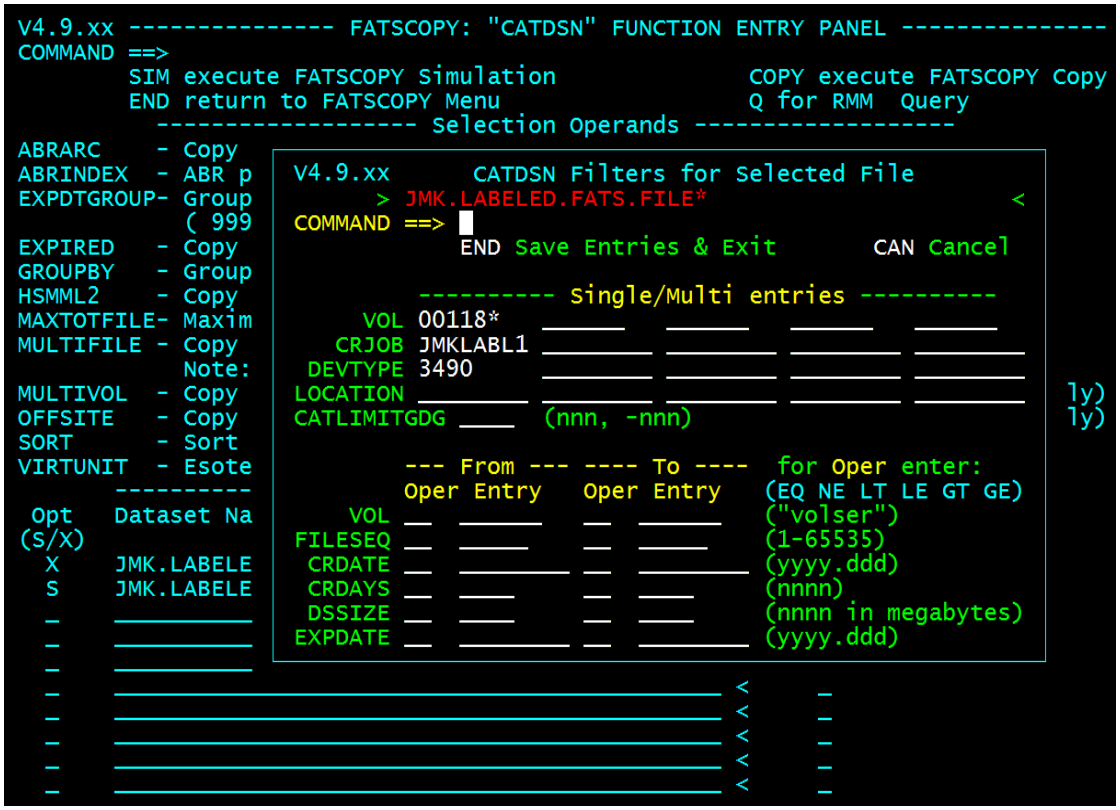
CATDSN Function Entry Panel

The FATSCOPY CATDSN Function Entry Panel provides many of the same options as the ALLDSN Function Entry Panel, but it is used to select and exclude data sets by data set name from the catalog. Under the Opt heading, you enter S to select data sets or X to exclude data sets that match the selection criteria. **Note that EXCLUDE statements must precede the SELECT statements to which they will be applied. An EXCLUDE statement applies to all following SELECT statements.**


V4.9.xx --------------- FATSCOPY: "CATDSN" FUNCTION ENTRY P Row 1 to 10 of 100
COMMAND ==>
SIM execute FATSCOPY Simulation
COPY execute FATSCOPY Copy
END return to FATSCOPY Menu
Q for RMM Query
Selection Operands
ABRARC - Copy ABR Archive or Application Backup files... NO (Yes,No)
ABRINDEX - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7 (0-999)
(999 means stack all files together regardless of exp date)
EXPIRED - Copy expired datasets (according to Tape Mgmt). NO (Yes,No)
GROUPBY - Group output by Expdate or by Volset ..... EXPDATE
HSMML2 - Copy HSM Migration Level 2 Volumes..... NO (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000 (1-65535)
MULTIFILE - Copy multi-file input tapes..... YES (Yes,No)
Note: specify Yes if input has more than 1 file
MULTIVOL - Copy multi-volume file datasets..... YES (Yes,No,Only)
OFFSITE - Copy offsite (vaulted) tape volumes..... NO (Yes,No,Only)
SORT - Sort by expiration date group and volser..... YES (Yes,No)
VIRTUNIT - Esoteric name for virtual tapes.....
Select/Exclude Datasets
Opt (S/X)
Dataset Name
Additional Limits (Y)
Have
x JMK.LABELED.FATS.FILE1*
S JMK.LABELED.FATS.FILE*
Y

Specifying Data Set Names to be Selected or Excluded

The example above shows how to code the selection criteria. Every time you add a data set name to the panel and hit ENTER, an additional blank line will be displayed, unless you enter a 'Y' under the Selection Criteria heading, indicating that you want to subject any data sets selected from the catalog to additional criteria, such as VOLSER, creation date, expiration date, etc. Coding data set masks is documented in Section 32.5 in this manual. In the example above, any data set that matches the first mask will be excluded (X). The second mask indicates data sets to be selected (S). The Y under the Selection Criteria heading indicates that you want to subject any data set that matches the catalog mask to additional criteria. When there is a 'Y' under the Selection Criteria for the current and you hit ENTER, the following panel will be displayed.



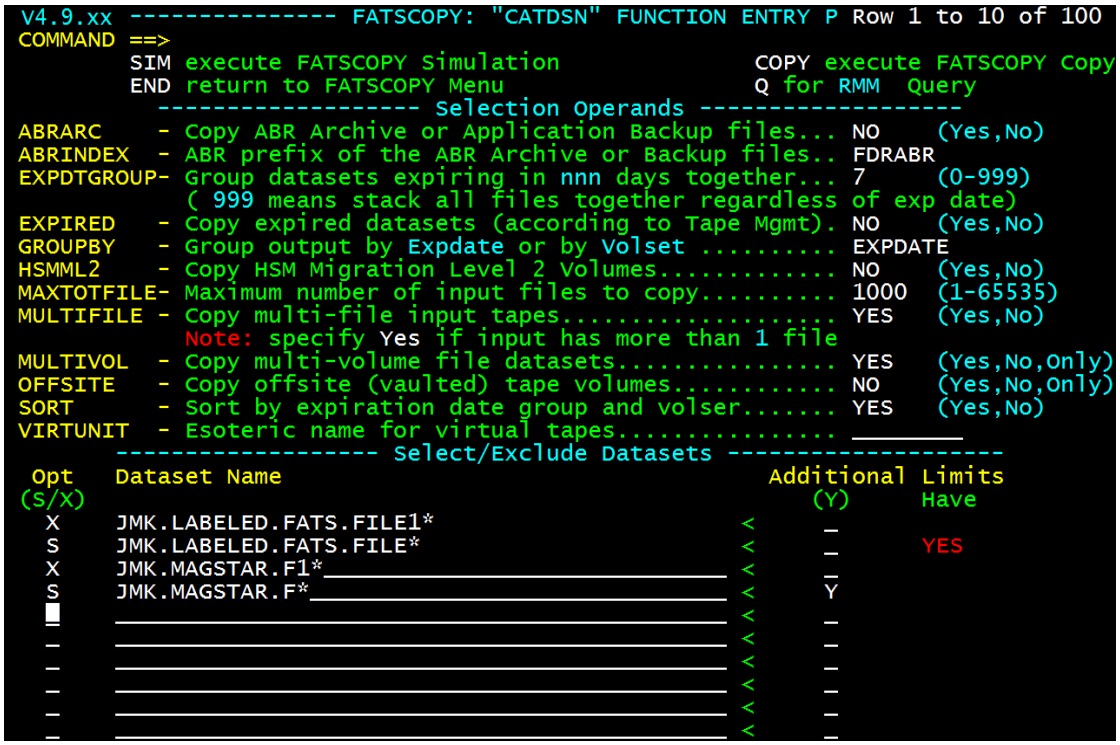
Entering Additional CATDSN Selection Criteria

In the example above, data sets matching the specified data set name mask will be selected only if they match the volume mask 00118*, were created by job JMKLABL1, ***and*** are cataloged to a 3490 volume. These selection criteria will apply ***only*** to data sets that match the DSN mask JMK.LABELED.FATS.FILE*. When you hit the END key, you will be returned to the previous panel, displayed below.

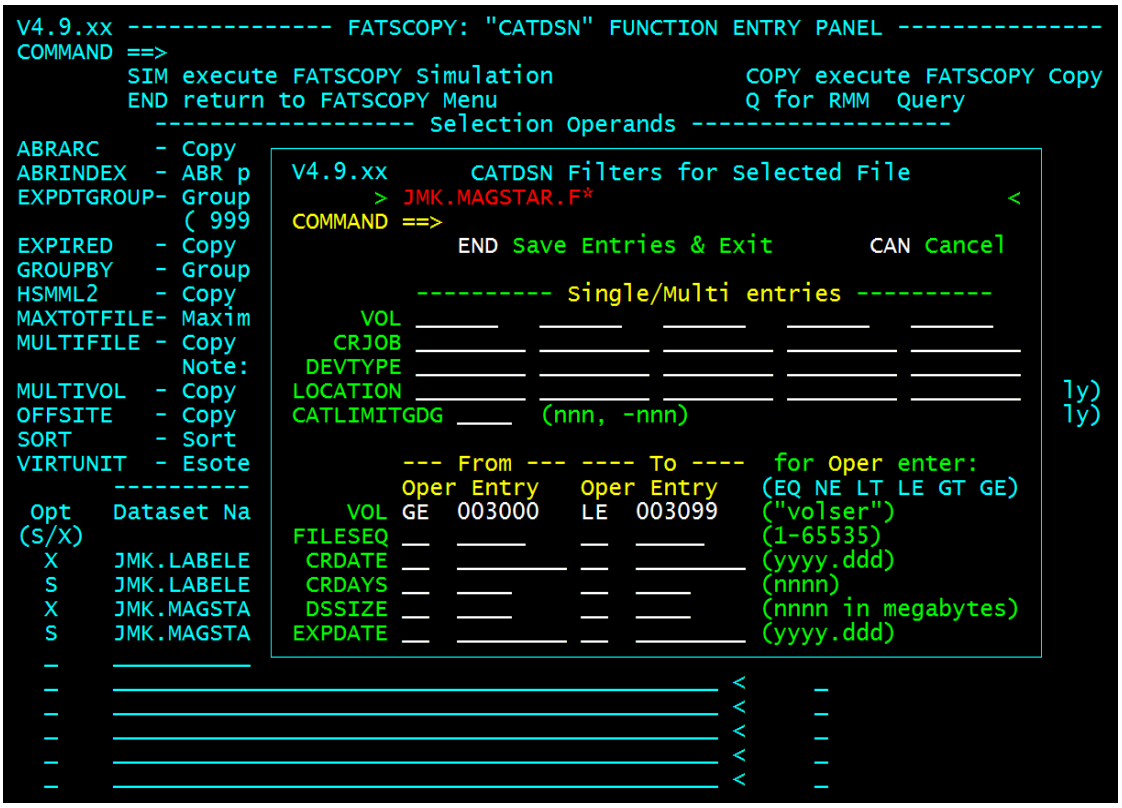
V4.9.xx --------------- FATSCOPY: "CATDSN" FUNCTION ENTRY P Row 1 to 10 of 100
COMMAND ==>
SIM execute FATSCOPY Simulation
COPY execute FATSCOPY Copy
END return to FATSCOPY Menu
Q for RMM Query
Selection Operands
ABRARC - Copy ABR Archive or Application Backup files... NO (Yes,No)
ABRINDEX - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7 (0-999)
(999 means stack all files together regardless of exp date)
EXPIRED - Copy expired datasets (according to Tape Mgmt). NO (Yes,No)
GROUPBY - Group output by Expdate or by Volset ..... EXPDATE
HSMML2 - Copy HSM Migration Level 2 Volumes..... NO (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000 (1-65535)
MULTIFILE - Copy multi-file input tapes..... YES (Yes,No)
Note: specify Yes if input has more than 1 file
MULTIVOL - Copy multi-volume file datasets..... YES (Yes,No,Only)
OFFSITE - Copy offsite (vaulted) tape volumes..... NO (Yes,No,Only)
SORT - Sort by expiration date group and volser..... YES (Yes,No)
VIRTUNIT - Esoteric name for virtual tapes.....
Select/Exclude Datasets
Opt (S/X)
Dataset Name
Additional Limits (Y)
Have
X JMK.LABELED.FATS.FILE1*
S JMK.LABELED.FATS.FILE*
YES

Adding Additional Data Sets and Selection Criteria

The 'YES' under the HAVE heading for DSN mask JMK.LABELED.FATS.FILE* indicates that additional selection criteria exist for that mask. If you fill this display and want to add additional data sets to be selected, press the DOWN key (usually PF8) or ENTER key to display additional lines where more data sets can be entered. You can press the UP key (usually PF7) to see previous lines that you have entered.


V4.9.xx --------------- FATSCOPY: "CATDSN" FUNCTION ENTRY P Row 1 to 10 of 100
COMMAND ==>
SIM execute FATSCOPY Simulation
COPY execute FATSCOPY Copy
END return to FATSCOPY Menu
Q for RMM Query
Selection Operands
ABRARC - Copy ABR Archive or Application Backup files... NO (Yes,No)
ABRINDEX - ABR prefix of the ABR Archive or Backup files.. FDRABR
EXPDTGROUP- Group datasets expiring in nnn days together... 7 (0-999)
(999 means stack all files together regardless of exp date)
EXPIRED - Copy expired datasets (according to Tape Mgmt). NO (Yes,No)
GROUPBY - Group output by Expdate or by Volset ..... EXPDATE
HSMML2 - Copy HSM Migration Level 2 Volumes..... NO (Yes,No)
MAXTOTFILE- Maximum number of input files to copy..... 1000 (1-65535)
MULTIFILE - Copy multi-file input tapes..... YES (Yes,No)
Note: specify Yes if input has more than 1 file
MULTIVOL - Copy multi-volume file datasets..... YES (Yes,No,Only)
OFFSITE - Copy offsite (vaulted) tape volumes..... NO (Yes,No,Only)
SORT - Sort by expiration date group and volser..... YES (Yes,No)
VIRTUNIT - Esoteric name for virtual tapes..... _____
Select/Exclude Datasets
Opt (S/X)
Dataset Name
Additional Limits (Y)
Have
X JMK.LABELED.FATS.FILE1*
S JMK.LABELED.FATS.FILE* YES
X JMK.MAGSTAR.F1*
S JMK.MAGSTAR.F* Y

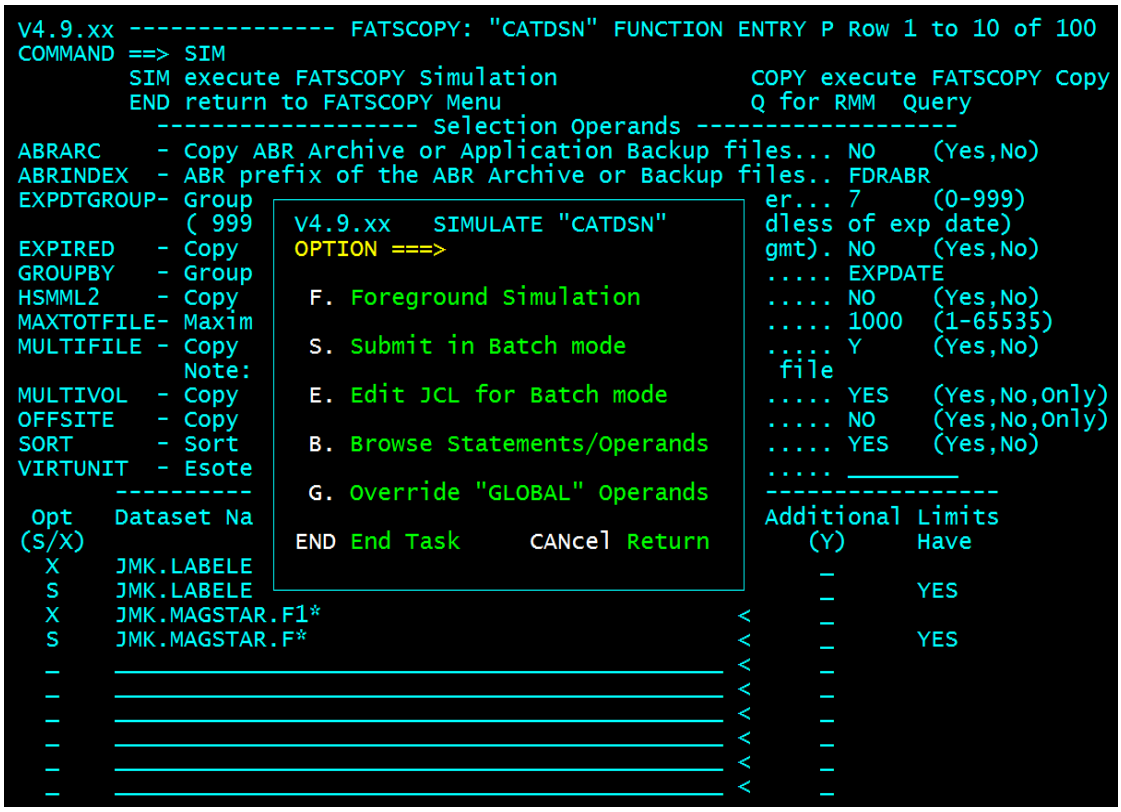
In the example above, a data set mask has been added to the list with a request to specify additional criteria.



CATDSN Filters – Volume Range

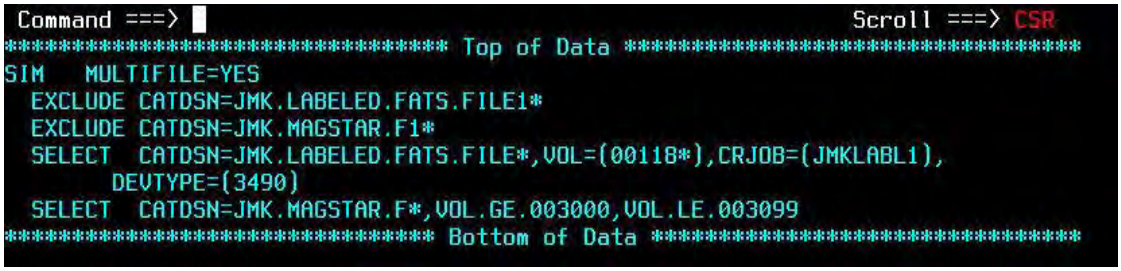
In the example above, a range of volumes has been added to the pop-up panel to limit the data sets selected by mask JMK.MAGSTAR.F* to those data sets that reside on volumes 003000 through 003099. Hitting the END key will return you to the previous panel.

When you are done adding all the masks to the panel, enter SIM on the command line, then END, to display a pop-up panel.



Simulation Options

Entering Option B in the above panel allows you to view the control statements that were generated. The control statements appear below.



Simulate Control Statements

After reviewing the control statements, hit the END key to return to the previous panel where you can choose to run an online simulation, submit a batch job, or other options. Selecting option 'S' to submit a batch job will produce the output shown on the next page.

CATDSN SAMPLE OUTPUT

```
FATSCOPY  INNOVATION TAPE COPY UTILITY   VER 4.9.xx       -- INNOVATION DATA PROCESSING  AUTHORIZED   2/22/2012 PAGE   1

FATS070 CONTROL CARD TABLE SIZE IS  32768 BYTES

FATSCOPY CONTROL CARDS

   1--  SIM     CHECKPT,MULTIFILE=YES
   2--  EXCLUDE CATDSN=JMK.LABELED.FATS.FILE1*
   3--  EXCLUDE CATDSN=JMK.MAGSTAR.F1*
   4--  SELECT  CATDSN=JMK.LABELED.FATS.FILE*,VOL=(00118*),CRJOB=(JMKLABL1),
                DEVTYPE=(3490)                                          ***CONTINUATION CARD***
FATS076 TAPE MANAGEMENT RMM
FATS346  FILTER SELECTED      10 ENTRIES IN     0.23 SECS
   5--  SELECT  CATDSN=JMK.MAGSTAR.F*,VOL.GE.003000,VOL.LE.003099
FATS346  FILTER SELECTED      25 ENTRIES IN     0.34 SECS
FATS120 **DSN=JMK.MAGSTAR.F2                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F3                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F4                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F5                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F6                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F7                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F8                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F9                      ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F20                     ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F21                     ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F22                     ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F23                     ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F24                     ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.MAGSTAR.F25                     ON VOLUME=003011 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.LABELED.FATS.FILE3              ON VOLUME=001186 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.LABELED.FATS.FILE4              ON VOLUME=001186 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.LABELED.FATS.FILE5              ON VOLUME=001186 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.LABELED.FATS.FILE7              ON VOLUME=001186 WAS SELECTED FOR COPYING.
FATS120 **DSN=JMK.LABELED.FATS.FILE8              ON VOLUME=001186 WAS SELECTED FOR COPYING.
FATS122 **FATSCOPY SIMULATE SUCCESSFULLY COMPLETED.      19 DATA SETS WERE SELECTED FOR COPYING.
```

FATSCOPY SYSPRINT Listing for CATDSN

The report above shows the control statements used as input. The CHECKPT parameter has been automatically inserted so that a subsequent FATSCOPY job can copy the data sets selected by this simulation. The FATS346 message shows the number of entries that matched the selection criteria for each CATDSN mask. The FATS122 message shows the total number of data sets that were selected for copying.

```
                                  FATSCOPY DATA SET COPY/SIMULATION REPORT                2/22/2012 PAGE      1

 DATA SET NAME                            VOLSER FSEQ  VSEQ  EXDATE  GROUP FUNC O/P EXP O/P VOL I/P  TMINFO OSEQ  RC
 ---------------------------------------- ------ ----- ----  -------- ----- ---- -------- ------- ---- ------ ----- ----
 JMK.MAGSTAR.F2                           003011     2    1  2015093     1 SIM  2015093         KEEP COPY       1 0000
 JMK.MAGSTAR.F3                           003011     3    1  2015093     1 SIM  2015093         KEEP COPY       2 0000
 JMK.MAGSTAR.F4                           003011     4    1  2015093     1 SIM  2015093         KEEP COPY       3 0000
 JMK.MAGSTAR.F5                           003011     5    1  2015093     1 SIM  2015093         KEEP COPY       4 0000
 JMK.MAGSTAR.F6                           003011     6    1  2015093     1 SIM  2015093         KEEP COPY       5 0000
 JMK.MAGSTAR.F7                           003011     7    1  2015093     1 SIM  2015093         KEEP COPY       6 0000
 JMK.MAGSTAR.F8                           003011     8    1  2015093     1 SIM  2015093         KEEP COPY       7 0000
 JMK.MAGSTAR.F9                           003011     9    1  2015093     1 SIM  2015093         KEEP COPY       8 0000
 JMK.MAGSTAR.F20                          003011    20    1  2015093     1 SIM  2015093         KEEP COPY       9 0000
 JMK.MAGSTAR.F21                          003011    21    1  2015093     1 SIM  2015093         KEEP COPY      10 0000
 JMK.MAGSTAR.F22                          003011    22    1  2015093     1 SIM  2015093         KEEP COPY      11 0000
 JMK.MAGSTAR.F23                          003011    23    1  2015093     1 SIM  2015093         KEEP COPY      12 0000
 JMK.MAGSTAR.F24                          003011    24    1  2015093     1 SIM  2015093         KEEP COPY      13 0000
 JMK.MAGSTAR.F25                          003011    25    1  2015093     1 SIM  2015093         KEEP COPY      14 0000

 JMK.LABELED.FATS.FILE3                   001186     3    1  2015106     2 SIM  2015106         KEEP COPY       1 0000
 JMK.LABELED.FATS.FILE4                   001186     4    1  2015106     2 SIM  2015106         KEEP COPY       2 0000
 JMK.LABELED.FATS.FILE5                   001186     5    1  2015106     2 SIM  2015106         KEEP COPY       3 0000
 JMK.LABELED.FATS.FILE7                   001186     7    1  2015106     2 SIM  2015106         KEEP COPY       4 0000
 JMK.LABELED.FATS.FILE8                   001186     8    1  2015106     2 SIM  2015106         KEEP COPY       5 0000


*** NUMBER OF OUTPUT TAPES REQUIRED:
**** NON-COMPRESSED OUTPUT:            2
**** COMPRESSED OUTPUT (3:1 RATIO):    2

 INPUT TAPES REQUIRED:
 001186  003011
```

FATSCOPY COPY/SIMULATION Report

The display above shows the data sets selected for copying. Data sets with the same group number will be stacked together. The report also shows the expiration dates that will be assigned to the output, whether the input will be kept or scratched, and whether tape management information will be propagated from the input to the output. In addition, it shows the input tapes and the approximate number of compressed or non-compressed output tapes needed to run the batch job.

REVIEW SIMULATION RESULTS SUBMITTED IN BATCH

If you submitted a Batch Simulation run from either the ALLDSN or the CATDSN Function Entry Panel, you can review the results of the batch job when it completes by selecting Option R from the FATSCOPY Selection Menu.

The following panel will be displayed:

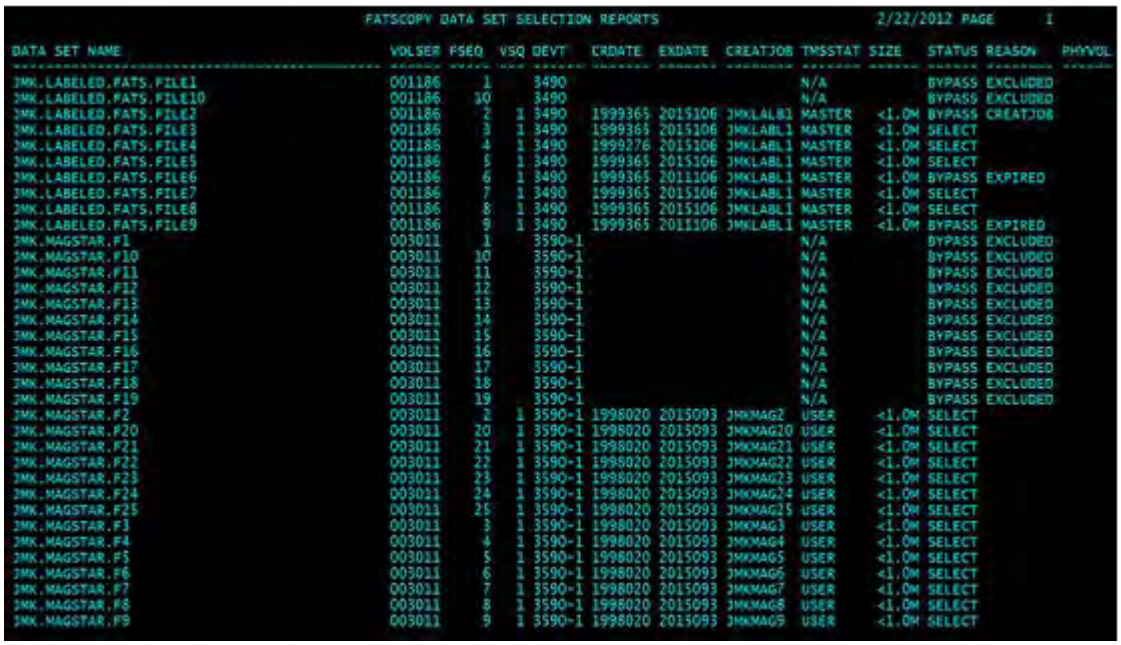
```
V4.9.xx ---------------- FATSCOPY - SIMULATION RUN DATASETS LIS Row 1 to 3 of 3
COMMAND ===>


Enter: S to Select for copy,  D to Delete the dataset
   ==> Alter Hi-Level Qualifier for the dataset search: MET0
       (You may enter * at end of Hi-level qualifier as a filter)

 S/D    Dataset Name                                  Created       Tracks
------------------------------------------------------------------------------
  _     MET0.FTCDSNDS.D2012045.T150053.MET0C          2012/02/14        15
  _     MET0.FTCDSNDS.D2012045.T160419.MET0C          2012/02/14        15
  _     MET0.FTCDSNDS.D2012059.T164038.MET0B          2012/02/28        15
****************************** Bottom of data ********************************
```

This shows that there are results from 3 batch simulation runs. If you have submitted jobs using a different high level qualifier, you can change the high level qualifier to display other data sets.

If you enter S to the left of a data set name, the FATSCOPY Simulation Review panel for that simulation run (shown in the FATSCOPY Online Simulation section on page 34-24) will be displayed. From that panel, you can

- View the Selection Report (option R)
- View a list of the selected data sets (option D)
- View a list of the selected volumes (option V)
- View any messages or errors generated during the simulation (option M)
- Submit a batch job to copy the selected data sets (option P)

Selecting option R on the Simulation Review panel for the above example results in the following display:



FATSCOPY DATA SET SELECTION REPORTS 2/22/2012 PAGE 1

DATA SET NAME	VOLSER	FSEQ	VSQ	DEVT	CRDATE	EXDATE	CREATJOB	TMSSTAT	SIZE	STATUS	REASON	PHYVOL
JMK.LABELED.FATS.FILE1	001186	1		3490				N/A		BYPASS	EXCLUDED	
JMK.LABELED.FATS.FILE10	001186	10		3490				N/A		BYPASS	EXCLUDED	
JMK.LABELED.FATS.FILE2	001186	2	1	3490	1999365	2015106	JMKLAL01	MASTER	<1.0M	BYPASS	CREATJOB	
JMK.LABELED.FATS.FILE3	001186	3	1	3490	1999365	2015106	JMKLABL1	MASTER	<1.0M	SELECT		
JMK.LABELED.FATS.FILE4	001186	4	1	3490	1999276	2015106	JMKLABL1	MASTER	<1.0M	SELECT		
JMK.LABELED.FATS.FILE5	001186	5	1	3490	1999365	2015106	JMKLABL1	MASTER	<1.0M	SELECT		
JMK.LABELED.FATS.FILE6	001186	6	1	3490	1999365	2011106	JMKLABL1	MASTER	<1.0M	BYPASS	EXPIRED	
JMK.LABELED.FATS.FILE7	001186	7	1	3490	1999365	2015106	JMKLABL1	MASTER	<1.0M	SELECT		
JMK.LABELED.FATS.FILE8	001186	8	1	3490	1999365	2015106	JMKLABL1	MASTER	<1.0M	SELECT		
JMK.LABELED.FATS.FILE9	001186	9	1	3490	1999365	2011106	JMKLABL1	MASTER	<1.0M	BYPASS	EXPIRED	
JMK.MAGSTAR.F1	003011	1		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F10	003011	10		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F11	003011	11		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F12	003011	12		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F13	003011	13		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F14	003011	14		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F15	003011	15		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F16	003011	16		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F17	003011	17		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F18	003011	18		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F19	003011	19		3590-1				N/A		BYPASS	EXCLUDED	
JMK.MAGSTAR.F2	003011	2	1	3590-1	1998020	2015093	JMKMAG2	USER	<1.0M	SELECT		
JMK.MAGSTAR.F20	003011	20	1	3590-1	1998020	2015093	JMKMAG20	USER	<1.0M	SELECT		
JMK.MAGSTAR.F21	003011	21	1	3590-1	1998020	2015093	JMKMAG21	USER	<1.0M	SELECT		
JMK.MAGSTAR.F22	003011	22	1	3590-1	1998020	2015093	JMKMAG22	USER	<1.0M	SELECT		
JMK.MAGSTAR.F23	003011	23	1	3590-1	1998020	2015093	JMKMAG23	USER	<1.0M	SELECT		
JMK.MAGSTAR.F24	003011	24	1	3590-1	1998020	2015093	JMKMAG24	USER	<1.0M	SELECT		
JMK.MAGSTAR.F25	003011	25	1	3590-1	1998020	2015093	JMKMAG25	USER	<1.0M	SELECT		
JMK.MAGSTAR.F3	003011	3	1	3590-1	1998020	2015093	JMKMAG3	USER	<1.0M	SELECT		
JMK.MAGSTAR.F4	003011	4	1	3590-1	1998020	2015093	JMKMAG4	USER	<1.0M	SELECT		
JMK.MAGSTAR.F5	003011	5	1	3590-1	1998020	2015093	JMKMAG5	USER	<1.0M	SELECT		
JMK.MAGSTAR.F6	003011	6	1	3590-1	1998020	2015093	JMKMAG6	USER	<1.0M	SELECT		
JMK.MAGSTAR.F7	003011	7	1	3590-1	1998020	2015093	JMKMAG7	USER	<1.0M	SELECT		
JMK.MAGSTAR.F8	003011	8	1	3590-1	1998020	2015093	JMKMAG8	USER	<1.0M	SELECT		
JMK.MAGSTAR.F9	003011	9	1	3590-1	1998020	2015093	JMKMAG9	USER	<1.0M	SELECT		

Data Set Selection Report

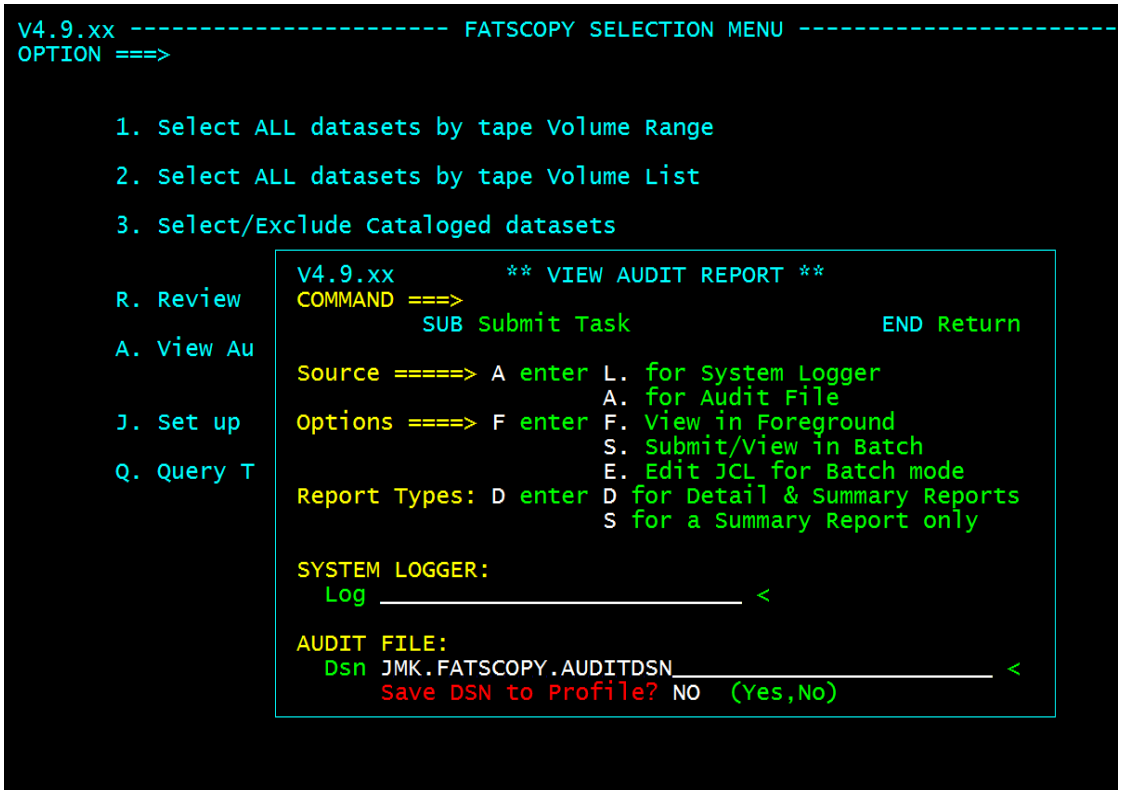
The above reports shows all the data sets that were selected from the catalog and whether they passed the tests for additional criteria. If a data set matched the DSN and volume masks, but were bypassed for some other reason, BYPASS will appear under the STATUS heading and an explanation of why the data set was bypassed will appear under the REASON heading. Some of the REASON indicators which may be displayed are:

- **NOTMINFO –** No tape management information for volume.
- **INV DSN –** Invalid data set name.
- **MISMATCH –** The data set name and location recorded in tape management does not match the name and location recorded in the system catalog.
- **MULTIFIL –** The data set was on a multifile volume and MULTIFILE=NO was used.
- **EXCLUDED –** The data set matched one of the criteria on an EXCLUDE statement.
- **EDM –** The data set is controlled by an external data manager.
- **DUPLICAT –** The data set was already selected by a previous SELECT statement.

In this report, 13 data sets were excluded by a EXCLUDE CATDSN statement, one was bypassed because it didn't match the creating job specified on the SELECT statement, and 2 were bypassed because they were expired. Although no expiration date criteria were entered, FATSCOPY uses a default of EXPIRED=NO for CATDSN and won't select any expired data sets.

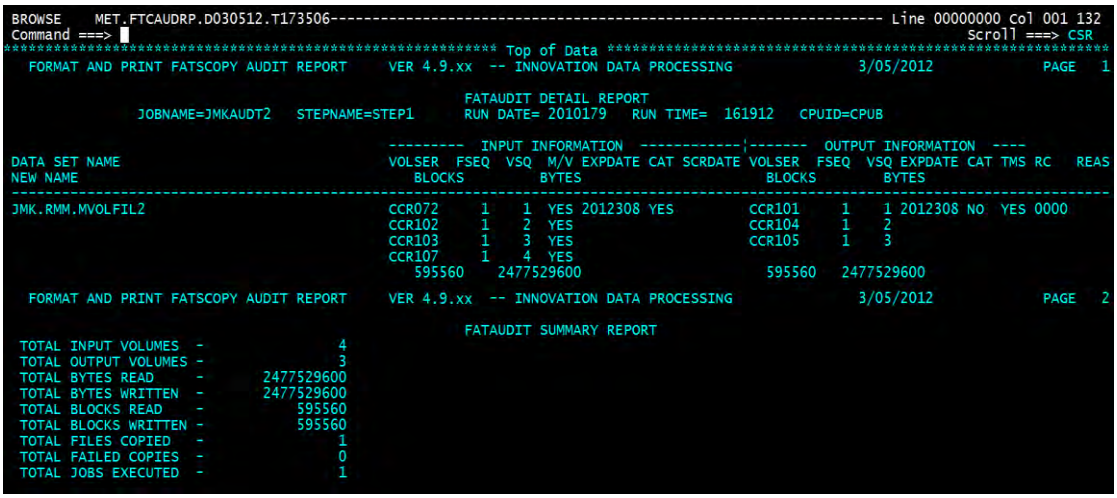
VIEW AUDIT REPORT

Selecting Option 'A' from the FATSCOPY Selection Menu Panel will display a Generate Audit Report pop-up panel.


V4.9.xx ---------- FATSCOPY SELECTION MENU ----------
OPTION ===>
1. Select ALL datasets by tape Volume Range
2. Select ALL datasets by tape Volume List
3. Select/Exclude Cataloged datasets
R. Review
A. View Au
J. Set up
Q. Query T
V4.9.xx ** VIEW AUDIT REPORT **
COMMAND ===>
SUB Submit Task END Return
Source =====> A enter L. for System Logger
A. for Audit File
Options ====> F enter F. View in Foreground
S. Submit/View in Batch
E. Edit JCL for Batch mode
Report Types: D enter D for Detail & Summary Reports
S for a Summary Report only
SYSTEM LOGGER:
Log <
AUDIT FILE:
Dsn JMK.FATSCOPY.AUDITDSN <
Save DSN to Profile? NO (Yes,No)

In the panel above, 'D' has been entered for the Report Type, to produce both a Detail and a Summary Report. The report will be produced using the sequential audit data set JMK.FATSCOPY.AUDITDSN, and the report will be run in the foreground.

Note that selecting “L” for System Logger will give a valid results only if your system is set up by your system programmer to use the system logger to write an audit log. You must also provide the log stream name in the “Log” field. See page 32-35 for information on using the system logger to write audit records.



```
BROWSE    MET.FTCAUDRP.D030512.T173506---------------------------------------------------------- Line 00000000 Col 001 132
Command ===> _                                                                                   Scroll ===> CSR
******************************************************* Top of Data ********************************************************
  FORMAT AND PRINT FATSCOPY AUDIT REPORT     VER 4.9.xx  -- INNOVATION DATA PROCESSING              3/05/2012            PAGE   1

                                             FATAUDIT DETAIL REPORT
              JOBNAME=JMKAUDT2   STEPNAME=STEP1      RUN DATE= 2010179   RUN TIME=  161912   CPUID=CPUB

                                          ---------  INPUT INFORMATION  ------------¦-------  OUTPUT INFORMATION  ----
DATA SET NAME                             VOLSER  FSEQ  VSQ  M/V EXPDATE CAT SCRDATE VOLSER  FSEQ  VSQ EXPDATE CAT TMS RC   REAS
NEW NAME                                    BLOCKS         BYTES                    BLOCKS        BYTES
-------------------------------------------------------------------------------------------------------------------------------
JMK.RMM.MVOLFIL2                          CCR072     1    1  YES 2012308 YES       CCR101     1    1 2012308 NO  YES 0000
                                          CCR102     1    2  YES                   CCR104     1    2
                                          CCR103     1    3  YES                   CCR105     1    3
                                          CCR107     1    4  YES
                                            595560    2477529600                    595560   2477529600

  FORMAT AND PRINT FATSCOPY AUDIT REPORT     VER 4.9.xx  -- INNOVATION DATA PROCESSING              3/05/2012            PAGE   2

                                             FATAUDIT SUMMARY REPORT

  TOTAL INPUT VOLUMES  -                4
  TOTAL OUTPUT VOLUMES -                3
  TOTAL BYTES READ     -       2477529600
  TOTAL BYTES WRITTEN  -       2477529600
  TOTAL BLOCKS READ    -           595560
  TOTAL BLOCKS WRITTEN -           595560
  TOTAL FILES COPIED   -                1
  TOTAL FAILED COPIES  -                0
  TOTAL JOBS EXECUTED  -                1
```

FATSCOPY Sample Audit Report

The listing above shows the audit report generated by the panel. The Detail Report provides information about the FATSCOPY job that generated the audit records, including job name, step name, the system where the job ran, and the date and time the job ran. The report shows information for 1 data set. The first line in the report shows the data set name that was copied, the expiration date of the input, whether or not the input was cataloged, and the volume that contained the data set. It shows all of the same information for the output data set and also includes whether the tape management information was propagated from the input to the output under the TMS heading. The first line also shows the return code issued when the data set was copied. If a non-zero condition code was issued, additional information will appear under the REAS heading. The next lines show any additional input or output volumes that were read or written, followed by a line showing the total number of bytes read from the input and written to the output.

The second page contains the Summary Report, which provides a summary of the information about the jobs, data sets, and volumes in the Detail Report.

35.0 FATAUDIT FUNCTIONAL DESCRIPTION

WHAT IS FATAUDIT?

FATAUDIT is a program that generates reports using the audit records created by FATSCOPY. These records may be created by FATSCOPY when an AUDIT DD is present in the JCL, or by using the interface with the system logger.

The **FATAUDIT Detail Report** contains information for each data set copied, including the job name, the time and date the job was run, the data sets and volumes copied, the number of bytes and blocks copied, and the return code issued by FATSCOPY when the data set is copied.

The **FATAUDIT Summary Report** shows the total number of files copied, total input and output volumes, and the total blocks and bytes read from the input tapes and written to the output tapes.

The audit reports provide valuable tools for tracking the copy process and determining whether any data sets weren't successfully copied.

FATAUDIT can also copy audit data from the system logger log stream into a sequential data set for long-term retention, and merge separate audit disk/tape data sets into a single output data set.

35.1 FATAUDIT EXECUTION JCL

To execute FATAUDIT, the following JCL statements are used.

EXEC STATEMENT

The required EXEC statement specifies the FATAUDIT program name, region size, and PARMs to be passed to the program. The optional values that can be specified in the PARM field are:

- **DETAIL** – A report will be produced showing detailed information for each data set copied, as well as a Summary Report showing a summary for all the data sets copied.
- **SUMMARY** – A report will be produced showing the number of files copied, the number of bytes read and written, the number of input and output volumes, and the number of jobs that failed.
- **LOG=*streamname*** – Audit data will be read from the system logger log stream ***streamname***. Any AUDIT DD data set(s) will be ignored when LOG= is used.
- **PURGE** – The log stream will be emptied at the end of the FATAUDIT job. **This permanently deletes the audit data recorded in the log stream**. When PURGE is used, an operator response is required to confirm that you want to delete this data. **Innovation recommends using an AUDITOUT DD statement when PURGE is specified** to copy audit data to a sequential data set for your long-term records. PURGE is ignored if LOG= is not also specified.

If not overridden by specific values on the PARM, the default is to read data from the AUDIT DD data set and produce a Detail Report.

STEPLIB/JOBLIB DD STATEMENT

A STEPLIB or JOBLIB DD statement will be required if FATAUDIT has been link edited into a private library. It can be omitted if FATSCOPY is in a system library that can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). This **must** be an APF authorized library.

AUDITRPT DD STATEMENT

The formatted audit reports will be written to the required AUDITRPT DD. AUDITRPT is usually a SYSOUT data set with DCB attributes of RECFM=FBA,LRECL=133. If BLKSIZE= is specified, it must be a multiple of 133; otherwise, it will default to 133 for SYSOUT and 1330 for other devices.

AUDIT DD STATEMENT

The AUDIT DD is a disk or tape data set that was created by a FATSCOPY job that contains the audit records used to generate the FATSCOPY Audit Report. The data set has a RECFM=FB and an LRECL=220. The AUDIT DD statement is required unless you are reading log data from the system logger (LOG=*streamname* PARM on the EXEC statement), in which case the AUDIT DD statement will be ignored.

Multiple data sets may be concatenated in the AUDIT DD statement. This allows the results saved in several audit data sets to be displayed in a single report, and to be written out as a single merged data set if the AUDITOUT DD statement is also used.

AUDITOUT DD STATEMENT

The optional AUDITOUT DD is a disk or tape data set to which FATAUDIT will write out all of the audit records it has read in (from the system logger log stream, or from one or more AUDIT DD data sets). This is useful for saving the contents of the log stream, especially before purging the log stream, or to merge separate audit data sets into one single data set. The format of the AUDITOUT data set is the same as the AUDIT data set; it is not formatted as a user-readable report. The data set produced as an AUDITOUT data set in one FATAUDIT job can be used as an AUDIT data set by another FATAUDIT job or by a FATSCOPY job.

SYSABEND DD STATEMENT

SYSABEND requests an abend dump if major errors occur. SYSABEND is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by Innovation.

FATAUDIT EXAMPLE 1: USING A SEQUENTIAL AUDIT DATA SET

The following example will produce a Detail Report using the Audit data set JAT.FATSCOPY.AUDITDSN previously created by FATSCOPY.

```
//AUDITJOB  EXEC  PGM=FATAUDIT,REGION=0M,PARM='DETAIL'
//STEPLIB   DD    DSN=FATSYS.PROD.LOAD,DISP=SHR
//SYSABEND  DD    SYSOUT=*
//AUDITRPT  DD    SYSOUT=*
//AUDIT     DD    DSN=JAT.FATSCOPY.AUDITDSN,DISP=SHR
/*
```

FATAUDIT EXAMPLE 2: MERGING AUDIT DATA SETS

The following example will read in 3 sequential Audit data sets, display a Summary Report, and write out all of the audit data into a single AUDITOUT data set.

```
//AUDITMRG EXEC PGM=FATAUDIT,REGION=0M,
//          PARM='SUMMARY'
//STEPLIB  DD DSN=FATSYS.PROD.LOAD,DISP=SHR
//AUDIT    DD DSN=JAT.FATSCOPY.AUDIT.MON,DISP=SHR
//          DD DSN=JAT.FATSCOPY.AUDIT.TUES,DISP=SHR
//          DD DSN=JAT.FATSCOPY.AUDIT.WED,DISP=SHR
//AUDITOUT DD DSN=JAT.FATSCOPY.AUDIT.WEEK1,UNIT=SYSALLDA,
//             DISP=(NEW,CATLG),SPACE=(TRK,(5,5))
//SYSABEND DD SYSOUT=*
//AUDITRPT DD SYSOUT=*
/*
```

FATAUDIT EXAMPLE 3: READING SYSTEM LOGGER LOG STREAM

The following example reads in audit data from the system logger, using the FATSCOPY log stream named FATSLOGR.AUDITLOG. A Detail Report will be written.

```
//AUDITLOG EXEC PGM=FATAUDIT,REGION=0M,
//          PARM='LOG=FATSLOGR.AUDITLOG,DETAIL'
//STEPLIB  DD DSN=FATSYS.PROD.LOAD,DISP=SHR
//SYSABEND DD SYSOUT=*
//AUDITRPT DD SYSOUT=*
/*
```

FATAUDIT EXAMPLE 4: WRITING LOG STREAM DATA TO A DISK FILE, AND ERASING THE LOG STREAM

The following example reads in audit data from the system logger and writes a Summary Report. The data read from the log stream will be written to a disk file, and the log stream data will be erased after the operator confirms the request.

```
//AUDITPUR EXEC PGM=FATAUDIT,REGION=0M,
//          PARM='LOG=FATSLOGR.AUDITLOG,SUMMARY,PURGE'
//STEPLIB  DD DSN=FATSYS.PROD.LOAD,DISP=SHR
//AUDITOUT DD DSN=JAT.FATSCOPY.AUDIT.APRIL,UNIT=SYSALLDA,
//             DISP=(NEW,CATLG),SPACE=(TRK,(5,5))
//SYSABEND DD SYSOUT=*
//AUDITRPT DD SYSOUT=*
/*
```

SAMPLE FATAUDIT DETAIL REPORT

This is a sample of a small FATAUDIT Detail Report that was created from an Audit data set which was used by two FATSCOPY jobs:

```
  FORMAT AND PRINT FATSCOPY AUDIT REPORT     VER 4.9.28  -- INNOVATION DATA PROCESSING              08/03/2013             PAGE   1

                                          FATAUDIT DETAIL REPORT
              JOBNAME=JATAUDIT   STEPNAME=PYTHON1    RUN DATE= 2013216   RUN TIME=  135106   CPUID=CPUB

                                        ---------  INPUT INFORMATION  ------------¦-------  OUTPUT INFORMATION  ----
DATA SET NAME                           VOLSER  FSEQ  VSQ  M/V EXPDATE CAT SCRDATE VOLSER  FSEQ  VSQ EXPDATE CAT TMS RC   REAS
NEW NAME                                   BLOCKS         BYTES                       BLOCKS        BYTES
-------------------------------------------------------------------------------------------------------------------------------
JAT.EE.FATS.FILE1                       CCR009     1    1  NO  2013001 YES         CCR157     1    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE2                       CCR009     2    1  NO  2013002 YES         CCR157     2    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE3                       CCR009     3    1  NO  2013003 YES         CCR157     3    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE4                       CCR009     4    1  NO  2013004 YES         CCR157     4    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE5                       CCR009     5    1  NO  2013005 YES         CCR157     5    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE6                       CCR009     6    1  NO  2013006 YES         CCR157     6    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE7                       CCR009     7    1  NO  2013007 YES         CCR157     7    1 2013283 NO  YES 0000
                                                 0            0                              0            0
JAT.EE.FATS.FILE8                       CCR009     8    1  NO  2013008 YES         CCR157     8    1 2013283 NO  YES 0000
                                                 0            0                              0            0
JAT.EE.FATS.FILE9                       CCR009     9    1  NO  2013009 YES         CCR157     9    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320
JAT.EE.FATS.FILE10                      CCR009    10    1  NO  2013010 YES         CCR157    10    1 2013283 NO  YES 0000
                                                 2         24320                             2         24320

  FORMAT AND PRINT FATSCOPY AUDIT REPORT     VER 4.9.28  -- INNOVATION DATA PROCESSING              08/03/2013             PAGE   2

                                          FATAUDIT DETAIL REPORT
              JOBNAME=JATLARGE   STEPNAME=PENGUIN1   RUN DATE= 2013216   RUN TIME=  140304   CPUID=CPUB

                                        ---------  INPUT INFORMATION  ------------¦-------  OUTPUT INFORMATION  ----
DATA SET NAME                           VOLSER  FSEQ  VSQ  M/V EXPDATE CAT SCRDATE VOLSER  FSEQ  VSQ EXPDATE CAT TMS RC   REAS
NEW NAME                                   BLOCKS         BYTES                       BLOCKS        BYTES
-------------------------------------------------------------------------------------------------------------------------------
JAT.RMM.BLK65536                        CCR130     1    1  NO  2013178 YES         CCR160     1    1 2013283 NO  YES 0000
                                                30         1966080                          30         1966080
JAT.RMM.BLK65535                        CCR130     2    1  NO  2013178 YES         CCR160     2    1 2013283 NO  YES 0000
                                                20         1285000                          20         1285000

  FORMAT AND PRINT FATSCOPY AUDIT REPORT     VER 4.9.28  -- INNOVATION DATA PROCESSING              08/03/2013             PAGE   3

                                          FATAUDIT SUMMARY REPORT

 TOTAL INPUT VOLUMES  -               2
 TOTAL OUTPUT VOLUMES -               2
 TOTAL BYTES READ     -         3445640
 TOTAL BYTES WRITTEN  -         3445640
 TOTAL BLOCKS READ    -              66
 TOTAL BLOCKS WRITTEN -              66
 TOTAL FILES COPIED   -              12
 TOTAL FAILED COPIES  -               0
 TOTAL JOBS EXECUTED  -               2
```

40.0 MESSAGES AND CODES

40.1 RETURN CODES

FATS RETURN CODES

FATS will set the following completion or return codes:

0 Normal completion. No serious errors.

12 Serious errors have occurred and the RETCODE option has been specified (user abend U0888 is issued otherwise).

32 A trial version of the product is being executed and the trial period has expired. Contact Innovation if you wish to continue your testing or install the production version.

FATAR/ FATSCOPY RETURN CODES

FATAR and FATSCOPY will set the following completion or return codes. Note that FATAR only sets return codes 4 and 8 if the SETNZRC operand is specified. If it is omitted, FATAR will set return code 0 for those conditions.

0 Normal completion. No serious errors.

4 Unusual conditions occurred, including: unable to load PRINT or ASCII translate table, EOD (end of data) mark encountered, FATSCOPY called for a new output tape during LASTAPE processing. However, the function ran to completion.

8 Unusual conditions occurred, including: missing DD statements, TAPEIN block count incorrect, VBS spanning error, unable to deblock/reblock, control statement not referenced, unable to catalog or rename output data set, tape management errors and other conditions. The function may not have completed; check the messages.

12 Serious errors have occurred and the RETCODE option has been specified (user abend U0200 or U0888 is issued otherwise). FATSCOPY will also set return code 12 if dynamic allocation or OPEN of an input tape fails.

32 A trial version of the product is being executed and the trial period has expired. Contact Innovation if you wish to continue your testing or install the production version.

68 One of the FATSCOPY interface modules is out of sync. This will usually occur if FATSCOPY loads a back level version of FATRMM, FATTMS, or FATTLMS. If you recently installed a new release of FATSCOPY, verify that FATSCOPY isn't picking up a back level version of one of these modules from the linklist.

40.2 ABEND CODES

ABEND CODES FATS, FATAR, and FATSCOPY may issue the following user and system abend codes. Most abends will be accompanied by an explanatory message.

S0C3 A S0C3 abend in FATS, FATAR, or FATSCOPY occurs when an unexpected condition occurs to create a dump to help Innovation debug the problem. Save the dump and contact Innovation for assistance.

U0001 FATSCOPY was unable to obtain the amount of storage required for its internal tables. Add REGION=0M to your JOB or STEP statement and rerun the job. Contact Innovation for assistance if the problem persists.

U0047 The FATS, FATAR, or FATSCOPY program is not authorized. See message FATS112.

U0100 For FATAR, the TAPEIN DD statement is missing, TAPEIN or TAPEOUT is not assigned to a tape unit, or an open failure on TAPEIN or TAPEOUT. For FATSCOPY, only 1 volume was selected for copying and an open error occurred when FATSCOPY tried to open the volume.

U0200 (FATAR only) Unrecoverable I/O error on TAPEOUT. If the RETCODE parameter is specified, a return code of 12 is issued instead of this abend.

U0300 (FATSCOPY only) No data sets were selected. This abend may occur if the input contains more than 1 file and MULTIFILE=YES wasn't specified on the COPY statement. It may also occur if all of the data sets that would have been selected were marked as Offsite. Check the information under the REASON heading in the Data Set Selection Report for the reason that a data set was not selected. Some of the REASON indicators are:

- **NOTMINFO –** No tape management information for volume.
- **INV DSN –** Invalid data set name.
- **MISMATCH –** The data set name and location recorded in tape management does not match the name and location recorded in the system catalog.
- **MULTIFIL –** The data set was on a multifile volume and MULTIFILE=NO was used.
- **EXCLUDED –** The data set matched one of the criteria on an EXCLUDE statement.
- **EDM –** The data set is controlled by an external data manager.
- **DUPLICAT –** The data set was already selected by a previous SELECT statement.

U0401 Open error or I/O error on SYSIN.

U0402 Open error or I/O error on a PRINT data set.

U0403 The PARM passed to FATAUDIT was invalid.

U0404 FATAUDIT was unable to connect to the specified logger stream name. Check that the name is correct.

U0405 FATAUDIT was unable to read any data from the specified log stream. The stream may be empty.

U0406 FATAUDIT encountered an error reading data from the specified log stream.

U0407 FATAUDIT encountered an error processing the records from a log stream data block.

U0408 FATAUDIT was unable to properly disconnect from the log stream.

U0409 FATAUDIT was unable to connect to the log stream in write mode when PURGE was used.

U0410 FATAUDIT was unable to purge the log stream.

U0411 FATAUDIT was unable to open the AUDITOUT data set.

U0502 Error in FATS, FATAR, or FATSCOPY control statements.

U0888 Major errors on tape. For FATS, any condition causing abnormal completion of a FATS function (indicated by "ABNORM" in the FATS summary report). For FATAR and FATSCOPY, unrecoverable I/O errors on TAPEIN other than data checks, or MAXERR= or MAXTERR= values exceeded. If the RETCODE parameter is specified, a return code of 12 is issued instead of this abend.

U0913 Security for FATS (Section 11.8) or FATAR (Section 21.6) has been implemented and the user did not have authority to perform the requested function.

40.3 CONSOLE MESSAGES

CONSOLE MESSAGES

The following write-to-operator (WTO) and write-to-operator-with-reply (WTOR) messages may be issued by FATS, FATAR, or FATSCOPY. They will be written to route code 3 (tape pool console).

FATSW01 **REQUEST TO LABEL TAPE ON UNIT uuuu REPLY WITH SERIAL NUMBER**

Reason: The OPERATOR parameter has been specified or the SAVLAB function found that the tape was not a labeled tape.

Action: Respond with a 1-to-6 character volume serial. FATS will left justify it, blank fill it, and label the tape on tape unit "uuuu".

FATSW02 **REPLY EOJ/KEOJ/Kuuuu TO TERMINATE**

Reason: FATS will always issue this WTOR at the start of every FATS run to allow the operator to control and terminate FATS.

Action: It will not be necessary to reply to this message unless:

1) MULT was specified without MAXVOLN=;

2) It is desired to terminate FATS before its normal termination; or

3) A particular tape must be halted. Valid responses are:

EOJ -- FATS will terminate after all tapes that are currently in progress have completed. Any pending tape mounts must be satisfied.

KEOJ -- FATS will halt all tapes in progress, cancel any pending mounts, and terminate with a U0888 abend or return code 12.

Kuuuu -- Where "uuuu" is the 3 or 4-digit address of a tape drive allocated to FATS. Any mount or tape in progress on that drive will be halted. If appropriate, another tape may be requested on the drive. Other drives allocated to FATS will not be affected. If "uuuu" is not a drive in use by FATS, it will be ignored. The FATSW02 message will be reissued.

FATSW03 **FATS ANALYSIS OF TAPE ON UNIT=uuuu VOL=vvvvvv PERM ERRORS=ppppp TEMP ERRORS=ttttt**

Reason: The WTO parameter has been specified. This message will be issued at the end of processing for each tape, giving unit "uuuu", volume serial "vvvvvv", and the total number of permanent and temporary errors ("ppppp" and "ttttt"). If a tape has major errors causing it to be halted prematurely, "ppppp" will be "MAJOR".

FATSW04 **MOUNT NEXT TAPE ON UNIT=uuuu VOL=vvvvvv**

Reason: Issued by FATS to request mounting of another tape on tape unit "uuuu". The volume serial "vvvvvv" may be an explicit serial or "SCRTCH" as appropriate.

Action: Mount the requested tape on unit "uuuu".

FATSW05 **FATS HAS STOPPED ON DATA CHECK ON UNIT=uuuu REPLY CONT, TERM, NOSTOP**

Reason: The STOP or STOPNUM= parameters have been specified and a permanent data check has occurred. FATS stops to allow visual inspection of the tape at the point of the error.

Action: Inspect the tape on tape unit "uuuu", if desired, re-ready the tape, and reply:

CONT -- To continue certifying/verifying, stopping again on the next data check.
TERM -- To terminate processing of the tape.
NOSTOP -- To continue certifying/verifying without stopping on further data checks.

Note that if you remove the tape from the "READY" state to inspect it, you will probably be unable to re-ready it at the same spot and be forced to reply "TERM".

FATSW06 PLEASE CLEAN UNIT=uuuu THEN (RE)MOUNT VOL=vvvvvv

Reason: The MAXCLEAN= or ERRCLEAN= parameters were specified. If MAXCLEAN=n then "n" tapes have been processed on unit "uuuu" since the last cleaning. If ERRCLEAN=n then more than "n" errors have occurred on unit "uuuu" (the message will say REMOUNT).

Action: Clean tape unit "uuuu" and mount or remount the specified volume "vvvvvv".

FATSW07 INVALID RESPONSE TO TERMINATE MESSAGE -- RE-ENTER

Reason: The response to message FATSW02 was not "EOJ", "KEOJ", or "Kuuuu". FATSW02 will be reissued.

Action: Reply with a valid response for message FATSW02.

FATSW08 VOL=vvvvvv MOUNTED,DSN=d...d,EXPDT=yyddd
UNIT=uuuu REQUIRES VOL=xxxxxx-REPLY IGNORE, UNLOAD, TERM, SKIP
UNIT=uuuu DSN NOT EXPIRED -- REPLY IGNORE, TERM, SKIP

Reason: The VALIDATE= parameter was specified to validate input and/or output labeled tapes with LABEL=(,BLP). Either the wrong volume was mounted or the expiration date on an output volume was not reached. In the first line of the message, "uuuu" is the tape unit, "vvvvvv" is the actual volume mounted, "d...d" is the data set name read from the tape, and "yyddd" is the expiration date. One of the next two formats of the message will be issued ("xxxxxx" is the volume expected).

Action: Reply with:

IGNORE -- To ignore the error and accept the volume mounted.
UNLOAD -- To unload the volume and request mounting of another tape.
TERM -- To terminate FATS or FATAR.
SKIP -- To terminate this volume and go on to the next (FATS only).

FATSW11 FATAR FILE ffff OPERATION UNIT=uuuu,VOL=vvvvvv,BLOCKS=bbbbbb,PERM ERRS=ppppp,TEMP ERRS=ttttt

Reason: The WTO parameter has been specified. The message will be issued at the end of processing each input file, giving file number "ffff", unit "uuuu", volume serial "vvvvvv", input block count "bbbbbb", and the total number of permanent and temporary errors ("ppppp" and "ttttt") in that file. "Operation" will be "ANALYZED ON" (if TAPEOUT absent), "COPIED FROM" (if TAPEOUT present), or "VERIFIED ON" (if verifying TAPEOUT)

Action: None

FATSW12 FATAR FILE ffff OPERATION UNIT=uuuu,VOL=vvvvvv REPLY CONT, TERM, OR NOSTOP

Reason: The STOP parameter has been specified. The message will be issued at the end of processing each input file, giving file number "ffff", unit "uuuu", and volume serial "vvvvvv". "Operation" will be "ANALYZED ON" (if TAPEOUT is absent), or "COPIED FROM" (if TAPEOUT present).

Action: Reply with:

CONT -- To continue with the next input file, stopping again at the end of that file
TERM -- To terminate processing of TAPEIN
NOSTOP -- To continue with the next file without stopping at the end of further files

FATSW13 REPLY IS: TEXT

Reason: Automatic reply to tape management message.

FATSW14 INVALID MODIFY COMMAND GIVEN: text....

Reason: "text" was specified in MODIFY (F) id,text operator command. "id" is the FATS jobname for which the MODIFY command was intended. "text" is either "QUIT" or "Kuuuu" to terminate FATS when current tapes are done or to immediately terminate processing for tape unit "uuuu".

Action: None. Command is ignored.

FATSW15 **NO DSNTABLE DD. RESTART WILL FAIL. DO YOU WANT TO END JOB? YES/NO**

Reason: The operator entered an f jobname,stop console command to stop the FATSCOPY job when it finishes copying the current data set. FATSCOPY requires a DSNTABLE DD in the JCL to create the data set needed to restart the job.

Action: A reply of 'YES' will stop the job, which will need to be restarted from the beginning. A reply of 'NO' will allow the job to continue until it finishes copying all of the remaining data sets. **Innovation strongly recommends a reply of 'NO'.**

FATSW16 **OPEN OF AUDIT DATA SET FAILED. DO YOU WANT TO CONTINUE? REPLY YES/NO**

Reason: An AUDITRPT DD was found in the JCL, but FATSCOPY was unable to open the data set.

Action: A reply of 'YES' will allow the job to continue, but no audit records will be generated. A reply of 'NO' will end the job. Innovation strongly recommends a reply of 'NO'.

FATSW17 **FATSCOPY ABEND RECOVERY: STARTING**

FATSCOPY ABEND RECOVERY: ENDING

Reason: The operator entered a ***c jobname*** console command to stop the FATSCOPY job, or a FATSCOPY abend occurred. These messages indicate that the FATSCOPY routine to process the interruption was executed. If you specified a DSNTABLE DD statement in the job, these messages should be accompanied by a FATSW18 message.

Action: **Do not enter a CANCEL command while the recovery routine is running.**

FATSW18 **FATSCOPY ABEND RECOVERY: DSNTABLE DATASET WRITTEN**

Reason: The operator entered an ***c jobname*** console command to stop the FATSCOPY job, or a FATSCOPY abend occurred. A DSNTABLE DD was present in the JCL, so FATSCOPY was able to create the data set needed to restart the job from the point where the job was interrupted. A FATS189 print message will also be issued.

FATSW19 ****WARNING** NO AUDITOUT DD USED. LOGSTREAM DATA WILL *NOT* BE WRITTEN TO DISK.**

Reason: The PURGE parameter was used in a FATAUDIT job, but no AUDITOUT DD statement was used**. Information in the audit log stream created after the last time an AUDITOUT was written will be lost** if the log stream purge is done.

FATSW20 ****WARNING** PURGE PARM USED. ==>THIS WILL EMPT Y THE CONTENTS FROM THE LOG STREAM DATA SET<==**

DO YOU WANT TO PURGE THE LOG STREAM? REPLY YES/NO

Reason: The PURGE parameter was used in a FATAUDIT job. The operator must confirm that a purge is to be done. A purge **permanently empties** the audit information in the log stream. If an AUDITOUT DD statement is used, the data from the log stream will be written to the AUDITOUT dataset before the purge is done.

Action: Reply **YES** or **NO**.

FATSW21 **PURGE PARAMETER WILL BE IGNORED.**

Reason: The operator replied NO to the FATSW20 message.

Action: The log stream data will not be purged.

FATSW22 FATSCOPY UNABLE TO ESTABLISH ESTAE ROUTINE

Reason: FATSCOPY could not activate its error recovery routine.

Action: If an operator CANCEL is issued or an abend occurs, FATSCOPY will not be able to save the DSNTABLE data set which is normally used to restart an interrupted job.

FATSW89 -- xxxx -- TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN nnn DAYS.

-- xxxx -- TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRED ---------- PLEASE CONTACT INNOVATION.

Reason: "xxxx" (FATS,,FATAR, or FATSCOPY) trial will expire in "nnn" days or "xxxx" trial has expired. The version of FATS, FATAR, or FATSCOPY that you are running is a trial version and has expired or will expire shortly.

Action: If you have purchased FATS, FATAR, or FATSCOPY and have received the tape or electronic version, make sure that it was properly installed and the trial version has been deleted. Then rerun the job. If you have not yet purchased FATS, FATAR, or FATSCOPY, call Innovation.

FATSW99 SHOULD DUMP BE TAKEN? YES/NO

Reason: DUMP=ASK was specified on the FATAR or FATSCOPY statement and an ABEND that normally does not generate a dump was generated.

Action: The operator can reply **YES** or **NO** to indicate if an ABEND dump should be take. A SYSUDUMP or SYSABEND DD statement must be present in the step JCL.

40.4 PRINT MESSAGES

PRINT MESSAGES

FATS, FATAR, and FATSCOPY may issue the following messages to their data sets (SYSPRINT or SYSPRINx).

FATS001 **TAPEIN DD STATEMENT MISSING**

Reason: The DD statement for TAPEIN is missing from the job step JCL.
Action: FATAR is terminated with a U0100 abend. Correct the JCL and resubmit the job.

FATS002 **TAPEIN DEVICE IS NOT A TAPE UNIT**

Reason: The DD statement for TAPEIN does not reference a supported tape unit type.
Action: FATAR is terminated with a U0100 abend. Correct the JCL and resubmit the job.

FATS003 **TAPEIN COULD NOT BE OPENED**

Reason: The open of the input tape on TAPEIN failed.
Action: Program FATAR is terminated with a U0100 abend. Check the printout for operating system messages indicating the cause.

FATS005 **TAPEOUT DEVICE IS NOT A TAPE UNIT**

Reason: The DD statement for TAPEOUT does not reference a supported tape unit type.
Action: FATAR is terminated with a U0100 abend. Correct the JCL and resubmit the job.

FATS006 **TAPEOUT COULD NOT BE OPENED**

Reason: The open of the output tape on TAPEOUT failed.
Action: FATAR is terminated with a U0100 abend. Check the printout for operating system messages indicating the cause.

FATS010 **CONTROL CARD ERROR -- REASON x**

Reason: A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS010 message will be printed for each input statement (multiple errors will not be diagnosed). "x" is a character indicating the type of error detected:

1 – Open failed for the control statement data set (SYSIN or SYSINx).
2 – Control statement is completely blank.
3 – An invalid operation keyword was found on the control statement.
4 – More than one ANALYZE/COPY control statement was found.
5 – The control statement required at least one keyword parameter but did not contain one.
6 – A specific keyword parameter was required on the control statement but was not found; a required parenthesis was missing; a keyword was misspelled; or there is no tape management information because the tape management interface was not installed with FATZAPOP.
7 – An ANALYZE/COPY control statement was required but not found.
8 – T= on TABLE statement in error. Letter previously used.
9 – A numeric value for a keyword parameter exceeds the allowed maximum for that keyword.
A – A value for a keyword parameter was zero or non-numeric.
B – Continuation card required but not found.
C – B= keyword in error. Block range specified but not in ascending order.
D – D=, V=, or S= keyword in error. Length of data is more than 256 bytes or characters.
E – D=, V=, or S= keyword in error. Data field must contain at least 1 byte or character.
F – ENDAFTER= keyword error.
G – Control statement table is too small to contain all information from the control statements present. Increase the table size via the EXEC JCL statement PARM field.
H – Error in X' or C' data: no closing apostrophe, invalid hex digits, or an odd number of hex digits.
J – Invalid data string. Must start with X ' or C'
K – Error in T= keyword. On SCAN statement, must be A-Z and defined in a previous table statement (except for P and Z). On table statement, must be A-O or Q-Y and not previously used.
L – EXITNAME was not found, or length of EXITNAME is zero.
M – Data set name specified by NEWN= on a RENAME statement is invalid.
N – Data set name mask specified by NEWI= or CATDSN= on a SELECT, EXCLUDE or RENAME statement is invalid.
O – Invalid mask on SELECT statement or in HSMxxxMASK= operands.
P – Invalid volume list, volume mask, or volume range; conflicting selection parameters (such as DSN= and CATDSN=); or SELECT CATDSN and SELECT ALLDSN statements both present.
Q – Invalid value for ABRINDEX= operand.
R – Invalid value for OPCAP= operand.
S – Conflicting operands on the COPY statement.
T – Invalid esoteric name specified in the VIRTUNIT= or UNITIN= operands.
U – Conflicting RENAME statements: more than one RENAME applies to the same file.
V – LASTAPE data set name invalid.
W – Invalid keyword specified on EXCLUDE. EXCLUDE accepts only CATDSN, DSN, VOL, and DEVTYPE operands.
X – EXLASTPGM=, EXLASTJOB=, or THRESHOLD= specified, but is not valid for the tape management system in use.
Y – Invalid value specified in the GROUPBY= operand.
Z – Invalid value specified in the VIRTTYPE= operand, or PHYSVOL= or ALLPHYS= has been specified without also specifying VIRTTYPE=.

Action: FATAR/FATSCOPY will abend with a U0502 abend code after all control statements have been scanned.

FATS011 ****INVALID CONTINUATION CARD**

Reason: FATSCOPY detected an invalid continuation on an input statement.
Action: FATSCOPY will abend with a U0502 abend code. Correct the errors and re-execute.

FATS013 ONE OR MORE ERRORS DETECTED -- FATAR TERMINATED

Reason: Message FATS010 was issued for one or more control statement errors.

Action: FATAR will abend with a U0502 abend code. Correct the errors and re-execute.

FATS014 NO CONTROL CARD DATASET. DEFAULTS ASSUMED

Reason: No SYSIN DD statement for control statement input was present in the job step JCL; if FATAR was invoked by FATS, no SYSINx DD was present.

Action: All defaults are assumed for all ANALYZE statement parameters. If FATAR was invoked by FATS, any FATAR parameters specified in FATS statements will override these defaults.

FATS016 ANALYSIS TERMINATED DUE TO MAJOR ERROR

Reason: An unrecoverable hardware error has occurred on the input tape drive or channel. Preceding messages will give details of the error.

Action: FATAR is terminated with a U0888 abend or return code of 12. Contact Innovation if you need assistance in determining the cause.

FATS016 **COPY TERMINATED DUE TO MAJOR ERROR VOL=volser DSN=dsn

Reason: An unrecoverable hardware error has occurred on the input tape drive or channel. Preceding messages will give details of the error.

Action: FATSCOPY will terminate the current data set and select the next input data set.

FATS017 MAXIMUM ERRORS EXCEEDED -- FATAR TERMINATED

Reason: The number of unrecoverable data checks exceeded the MAXERR= parameter, or the number of recoverable data checks exceeded the MAXTERR= parameter.

Action: FATAR is terminated with a U0888 abend or return code of 12.

FATS017 MAXIMUM ERRORS EXCEEDED VOL=volser DSN=dsn COPY TERMINATED

Reason: The number of unrecoverable data checks exceeded the MAXERR= parameter, or the number of recoverable data checks exceeded the MAXTERR= parameter.

Action: FATSCOPY will terminate the current data set and select the next input data set.

FATS019 pgmname MUST BE IN AN AUTHORIZED LIBRARY. PROGRAM TERMINATED

Reason: FATAR and FATSCOPY must be executed as APF authorized programs. Either the program library containing FATAR or FATSCOPY was not marked as an APF authorized library on your z/OS system, or your STEPLIB or JOBLIB DD statement has nonauthorized libraries concatenated to the FATAR/FATSCOPY program library, or FATAR/FATSCOPY has been relinked as nonauthorized (AC=0).

Action: FATAR or FATSCOPY will terminate with a U0047 ABEND.

FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE

Reason: If labels are being processed by FATAR, two consecutive tape marks were read after a label file. If labels are not processed and NUMFILES=0 was specified or defaulted, two consecutive tape marks were read.

Action: Processing of the input tape is completed.

FATS020 ****COPY COMPLETE. ALL VOLUMES PROCESSED**

Reason: All of the volumes and data sets selected for copying have been processed. A zero return code indicates that FATSCOPY has copied all of the data sets, cataloged the output data sets, updated tape management and expired the inputs, if necessary. A non-zero condition code indicates errors occurred during the copy process, that may have resulted in some data sets not being copied, cataloged, or updated in tape management.

Action: Processing of the input data sets is completed.

FATS021 **ANALYSIS TERMINATED BY EOV LABELS DETECTED IN LABELS=NO MODE**

Reason: If labels are not being processed and NUMFILES=0 was specified or defaulted, FATAR has detected what appear to be IBM or ANSI EOV labels.

Action: Processing of the input tape is completed.

FATS022 **ANALYSIS TERMINATED AFTER SPECIFIED NUMBER OF FILES**

Reason: NUMFILES=n was specified, and "n" logical files have been read.

Action: Processing of the input tape is completed.

FATS023 **ANALYSIS TERMINATED BY ENDAFTER OPTION**

Reason: The file and block specified by the ENDAFTER= parameter was read.

Action: Processing of the input tape is completed.

FATS024 **ANALYSIS TERMINATED BY LOGICAL END OF FILE**

Reason: FATAR attempted to switch to a new input tape volume (due to EOV labels being read, or reaching end-of-file in LABELS=OS mode), but no more volumes were given.

Action: Processing of the input tape is completed.

FATS025 **ANALYSIS TERMINATED BY OPERATOR REQUEST**

Reason: The operator replied "TERM" to the FATSW12 message or FATSW08 message.

Action: Processing of the input tape is halted.

FATS026 **TAPEIN DATA SUCCESSFULLY COPIED TO TAPEOUT VOLUME VVVVVV**

Reason: The TAPEOUT DD statement was present, and reading of TAPEIN data completed successfully. All input data not "DROPPED" has been written to TAPEOUT. The last (or only) volume of TAPEOUT was "vvvvvv".

Action: None.

FATS027 **UNLABELED TAPEOUT REQUIRED MORE THAN ONE VOLUME -- VERIFY CANCELLED**

Reason: TAPEOUT is an unlabeled tape (LABEL=(,NL) or (,BLP)), the VERIFY parameter was specified, and TAPEOUT overflowed to more than one volume.

Action: FATAR cannot verify multi-volume unlabeled tapes. The VERIFY will not be performed.

FATS028 **ANALYSIS TERMINATED DUE TO INVALID LABEL**

Reason: Labels were being processed on TAPEIN, but a valid IBM standard or ISO/ANSI or user label was not read when one was expected.

Action: The invalid data will be printed, and FATAR will be terminated with a U0888 abend or return code 12

FATS030 EOT ON TAPEOUT BUT VOLUME SWITCH NOT ALLOWED -- FATAR TERMINATED

Reason: Labels are being processed on TAPEIN, the current file being copied contains user labels, and the SUPULAB parameter (suppress user labels) was not specified. User labels may be required with the EOV (end-of-volume) and header labels written when the file crosses volumes, but FATAR has no way to generate user labels acceptable to the user application.

Action: FATAR is terminated with a U0888 abend or return code 12.

FATS033 PERMANENT I/O ERROR ON TAPEOUT -- function TERMINATED

Reason: "**function**" may be FATAR or FATSCOPY, both of which use standard system error recovery on TAPEOUT. An uncorrectable I/O error occurred; messages will follow to detail the error that occurred.

Action: The function will be terminated with a U0200 abend or return code 12. Consult the messages and operating system messages for details of the error. Errors may be avoided by cleaning the TAPEOUT tape drive, and using an output tape certified by FATS or other means.

FATS034 NOISE BLOCK HAS BEEN READ VOL=volser DSN=dsn

Reason: A data check block too small to be considered a valid data block has been read; this will not occur on tape cartridge drives. The volume and dsn are displayed only by FATSCOPY.

Action: No recovery attempts will be made. The block will be immediately flagged as a permanent data check.

FATS035 INCOMPATIBLE TAPE FORMAT. DSN=dsname, VOL=volser

Reason: The tape drive has reported that it is not capable of reading the input tape. This may occur when a tape written in a certain format (such as 3490E or encrypted format) is mounted on a drive that is not capable of reading that format. This error may also be caused by an uninitialized tape. The volume and dsn are displayed only by FATSCOPY.

Action: FATAR will terminate with a U0888 abend. FATSCOPY will continue with the next input data set. Determine the actual format of the tape, and change the TAPEIN DD statement to allocate an appropriate tape drive.

FATS036 BLOCK GREATER THAN nnnnnn BYTES READ -- FATAR TERMINATED

Reason: A tape block longer than "nnnnnn" bytes (as specified by the BLKSIZE= parameter, default 64K) has been read. This form of the FATS036 message is produced only if an output tape is being written, i.e., a TAPEOUT DD statement is present.

Action: FATAR is terminated with a U0888 abend.

FATS036 ** WARNING ** BLOCK GREATER THAN nnnnnn BYTES READ -- BYTE COUNTS WILL BE INCORRECT

Reason: A tape block longer than "nnnnnn" bytes (as specified by the BLKSIZE= parameter, default 64K) has been read. This form of the FATS036 message is produced when no output tape (TAPEOUT DD statement) is present.

Action: The excess data is discarded, and FATAR continues with the next block. However, byte counts printed will not include the excess bytes.

FATS037 TAPE LABELS IN ISO/ANSI FORMAT -- TRANSLATED TO EBCDIC

Reason: FATAR has detected that the labels on TAPEIN are in ISO/ANSI format ("ANSI labels").

Action: The labels will be translated to EBCDIC for printing or processing. Also, the associated data file is assumed to be in ISCII/ASCII and will be translated to EBCDIC.

FATS038 DATA FILE IN ISCII/ASCII -- TRANSLATED TO EBCDIC

Reason: The current data file is assumed to be in ISCII or ASCII code. This occurs when DCB=OPTCD=Q is specified on the TAPEIN DD card, or when ISO/ANSI labels were detected on TAPEIN by FATAR.

Action: All data blocks in the file will be translated to EBCDIC for printing and processing.

FATS039 ****WARNING** BLOCK LENGTH GREATER THAN BLOCKSIZE OF bbbbbb DSN=dsname, VOL=volser**

Reason: A block was read whose length is greater than the blocksize "bbbbbb" of the current file (extracted from the header labels of the file, or given on the TAPEIN DD statement if labels are not processed). The volume and dsn are displayed only by FATSCOPY.

Action: None. Unless dropped or modified, the block will be written to TAPEOUT as read.

FATS040 **TAPEIN IS NOT LABELED -- LABELS=NO ASSUMED**

Reason: The LABELS=YES parameter was specified or defaulted, but FATAR did not find valid IBM or ISO/ANSI labels in the first file read.

Action: FATAR will proceed as if LABELS=NO were specified.

FATS042 **TAPEIN TRAILER BLOCK COUNT OF bbbbbb IS INCORRECT**

Reason: The EOF1 or EOV1 trailer label on a labeled TAPEIN file contained a block count "bbbbbb" that did not match the number of blocks actually read from the file by FATAR. The actual block count is found in the "end of file" message preceding this message.

Action: None.

FATS044 **END OF OUTPUT TAPE vvvvvv AFTER nnnnnnnnnn BLOCKS -- NEW VOLUME REQUESTED**

Reason: The end-of-tape reflector (tape indicate) was detected on TAPEOUT volume "vvvvvv" after the "nnnnnnnnnn"th block of the file being copied was written to the output tape.

Action: EOV labels are written on "vvvvvv" (if labeled), and a new output volume is requested.

FATS046 *** * I/O ERROR--> ECB=eeeeeeee ECB rc description**

Reason: This is the first of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the event control block (ECB, 4 bytes) in hexadecimal along with the description associated with the ECB reason code (1st byte of ECB).

Action: Other messages will indicate whether FATAR can continue or must terminate, depending on the nature of the error. Appropriate manufacturer's manuals should be consulted to determine the cause of the error. Contact Innovation if you need assistance.

FATS047 *** * I/O ERROR--> CSW=wwwwwwww wwwwwwww csw status desc**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the channel-status word. (CSW, 8 bytes) in hexadecimal along with the description associated with the unit and channel status bytes of the CSW (bytes 4 and 5).

Action: See message FATS046.

FATS048 *** * I/O ERROR--> CCW=cccccccc cccccccc ccw channel cmd**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the channel-command word (CCW, 8 bytes) in hexadecimal along with the channel command associated with the command code byte of the CCW (byte 0).

Action: See message FATS046.

FATS049 *** * I/O ERROR--> SENSE ssss EXTENDED SENSE ee...ee error description**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the sense bytes (ssss, 2 bytes) in hexadecimal. If available, the extended sense (up to 32 bytes) is displayed. If FATAR or FATSCOPY can determine the cause from the sense and extended sense, a text description of the error is displayed.

Action: See message FATS046.

FATS050 *** * I/O ERROR OCCURRED DSN=dsname,VOL=volser**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes identifying information for the data set being processed when the error occurred.

Action: See message FATS046.

FATS051 **RECORD rr BYTE bb LENGTH ll INVALID xxxx DATA FIELD**

Reason: A SCAN statement has detected data that does not meet the user-specified criteria. "rr" is the record number within the current block ("RECORD rr" will not appear if deblocking is not being done). "bb" is the location of the invalid field within the indicated record or block (relative to 1). "ll" is the length of the field. "xxxx" will be "ZONED", "PACKED" or "TYPE t" (where "t" is a table ID).

Action: None.

FATS052 **SCAN FIELD NOT IN RECORD -- SCAN TERMINATED FOR THIS RECORD**

Reason: A SCAN statement applied to a block or record, but the location to be scanned (P= parameter) was beyond the end of the block/record.

Action: SCAN not performed. Check the SCAN parameters.

FATS053 **BLOCK LENGTH ZERO -- BLOCK NOT KEPT**

Reason: A KEEP statement applied to an input block, but the length of the data read was zero (probably due to a data check).

Action: The block will not be written to TAPEOUT.

FATS054 **BLOCK LENGTH CHANGED WHILE IN LOGICAL RECORD MODE -- DEBLOCKING TERMINATED**

Reason: FATAR was deblocking input blocks into individual records, but a KEEP statement changed the length of a block.

Action: The block will not be deblocked.

FATS056 **RECORD TOO SHORT--NO DATA REPLACED**

Reason: A record or block is too short to contain the data specified on a REPLACE statement.

Action: REPLACE not performed. Check the REPLACE parameters.

FATS057 **DATA RECOVERED -- BLOCK NOT MODIFIED**

Reason: A Modification/Scan statement with the DCK parameter applied to a data block, but the block did not have a permanent data check.

Action: The Modification or SCAN will not be performed.

FATS058 **RECORD nnnnn HAS BEEN DROPPED**

Reason: In response to a DROP statement with a record number indicated, record "nnnnn" has been deleted from the current block.

Action: The block will be "compressed" to eliminate the deleted record. If the record format is variable, the block length will be updated.

FATS061 **RECFM/LRECL NOT GIVEN -- FILE ASSUMED UNBLOCKED**

Reason: The record format (RECFM) and logical record length (LRECL) for the current file are not available from either the TAPEIN DD statement or TAPEIN header labels.

Action: FATAR is unable to deblock the current file into logical records and will treat each block as a single record. If deblocking is desired, specify RECFM and LRECL on TAPEIN and re-execute FATAR.

FATS062 DEBLOCKING ANSI SPANNED RECORDS NOT SUPPORTED -- UNBLOCKED ASSUMED FOR THIS FILE

Reason: The TAPEIN DD statement or TAPEIN header labels specifies ANSI variable-length spanned records (record format DS or DBS). FATAR does not support deblocking of ANSI spanned records.

Action: Unblocked (record format U) assumed.

FATS063 BLOCK INVALID FOR RECFM/LRECL IN USE--DEBLOCKING TERMINATED

Reason: Blocks in the current file were being deblocked into logical records, but the current block cannot be deblocked using the record format and logical record length provided by the TAPEIN DD statement or TAPEIN header labels. For fixed length records, the block is not an exact multiple of the record length. For variable length records, the record descriptor words (RDW) may be invalid or indicate a length greater than the block size.

Action: The current block will be treated as unblocked. If the record format and length given are correct, the block is invalidly formatted; if not, specify the correct RECFM and LRECL on TAPEIN and re-execute FATAR.

FATS064 FILE NOT ELIGIBLE FOR REBLOCKING -- REBLOCKING SUPPRESSED FOR THIS FILE

Reason: REBLOCK= was specified, but the current file cannot be reblocked for one of the following reasons:

1) The record format is not fixed or variable.
2) The record format is variable spanned.
3) The blocksize is zero.
4) The logical record length is zero (if the record format is FB or VB).
5) The blocksize is greater than the MAXRBLK= parameter, if specified.
6) The blocksize of the input data set is equal to the blocksize requested on the REBLOCK operand. The output data set will not be reblocked.
7) The output blocksize is less than the input blocksize.

Action: The file will be copied without reblocking.

FATS065 RECORD LENGTH TOO LARGE FOR REBLOCKING -- REBLOCKING SUPPRESSED FOR THIS FILE

Reason: REBLOCK= was specified, but the logical record length of the current file is larger than the specified reblocking block size.

Action: The file will be copied without reblocking.

FATS066 THIS FILE WILL BE REBLOCKED TO A BLOCKSIZE OF nnnnn

Reason: REBLOCK= was specified, and the current file can be reblocked. The new block size "nnnnn" will be equal to the reblocking block size for variable length files, or the next lower multiple of the record length for fixed length files

Action: The file will be copied and reblocked.

FATS067 INPUT DEBLOCKING ERROR WHILE REBLOCKING -- FATAR TERMINATED

Reason: REBLOCK= was specified, and the current file was being reblocked, but an error occurred extracting logical records from the current input block; message FATS063 will precede this message.

Action: Since the labels on the output tape reflect the new blocksize, FATAR will be terminated with a U0888 abend or return code of 12. Specify the correct RECFM and LRECL on TAPEIN and re-execute FATAR.

FATS068 INPUT RECORD TOO LARGE WHILE REBLOCKING -- FATAR TERMINATED

Reason: REBLOCK= was specified, and the current file was being reblocked, but an input logical record was larger than the output reblocking blocksize.

Action: Since the labels on the output tape reflect the new blocksize, FATAR will be terminated with a U0888 abend or return code of 12. Specify a larger value for REBLOCK= and re-execute FATAR.

FATS069 RECORD nnnnn SPANNING ERROR -- xxxxx OF SEGMENT MISSING

Reason: The TAPEIN JCL or tape labels specifies variable spanned records for the current file, and FATAR has detected an incomplete spanned record. "xxxxx" will be "START" if the end of a record was found without a corresponding beginning, or "END" if a new record was found before a previous spanned record was complete. "nnnnn" is the record number within the current block where the error was found. If "xxxxx" is "END", the error is probably actually in the preceding block(s).

Action: An additional FATAR run may be required to print the appropriate blocks and determine which records are truly in error. Another run using FATAR REPLACE statements to fix the spanning flags, or DROP statements to delete the partial records, may be used to fix the tape.

FATS070 CONTROL CARD TABLE SIZE IS n BYTES

Reason: This message is always printed to document "n", the size of the FATAR control statement storage table. This size defaults to 32768 but can be increased by the SIZE= parameter in the FATAR EXEC JCL statement PARM=.

Action: If this FATAR execution receives message FATS010 reason G, increase the table size.

FATS071 TAPE BUFFER SIZE IS nnnnnn BYTES

Reason: This message is always printed to document the size of the FATAR/FATSCOPY TAPEIN read buffer. The default size is 262144 or 256K bytes but can be reduced by use of the BLKSIZE= parameter.

Action: Any blocks on TAPEIN exceeding "nnnnnn"+1 in length will cause an error. If FATAR is creating an output tape, it will terminate, otherwise it will continue (see message FATS036).

FATS072 TAPEIN DATA WILL BE COPIED TO TAPEOUT

Reason: The TAPEOUT DD statement is present.

Action: All files and data on TAPEIN will be copied to TAPEOUT unless "DROPPED" by data checks or FATAR control statements.

FATS073 **WARNING BACKUP OF ARCHIVE CONTROL FILE HAS SAME NAME AS ARCHIVE CONTROL FILE. OUTPUT DATA SET NOT CATALOGED.**

Reason: .ARCBACKUP=DSF was specified without a corresponding ARCB1DSN= keyword. When ARCB1DSN isn't specified, FATSCOPY creates the data set name for the backup of the Archive Control file by replacing the '.ARCHIVE' qualifier in the Archive Control File data set name with a '.ARCBKUP' qualifier. If FATSCOPY doesn't find a '.ARCHIVE' qualifier, it creates the backup with the same data set name as the Archive Control File and the output data set will not be cataloged.

Action: A return code '8' will be issued.

FATS074 ** WARNING ** CONTROL CARD NUMBER n WAS NEVER REFERENCED OR REPLACE-VERIFY NEVER SATISFIED

Reason: This message is issued at FATAR/FATSCOPY termination to warn of control statements that have never been acted upon. Either the file number or block number referenced on the statement was never encountered on TAPEIN, or, for a REPLACE statement with a V= or S= parameter, the VERIFY or SCAN never caused any data to be replaced.

Action: Review the control statements to be sure that they were coded correctly. Review the FATAR/FATSCOPY output to be sure that the desired files and blocks were processed.

FATS075 **SPECIAL EXPIRATION OF yydd1 DETECTED, TAPEOUT EXPDT=yydd2**

Reason: FATAR read labels containing an expiration date "yydd1" recognized as having a special purpose, i.e., 98000 or 99000.

Action: If copying to TAPEOUT, the output file will have expiration "yydd2" (98000 is normally changed to 00000, but some tape management systems will change this to a default retention).

FATS076 **TAPE MANAGEMENT text**

Reason: Displays information about the FATAR Tape Management interface. If no supported tape management system was detected or an interface error occurred, this is indicated. Otherwise it will identify the type and level of the tape management system, and information about the input tape extracted from that TMS.

Action: None, unless an error occurs. If necessary, contact Innovation Technical Support.

FATS077 **EOT ON TAPEOUT BUT FATAR WILL ATTEMPT TO WRITE nnn MORE BLOCKS**

Reason: During an image copy (LABELS=NO), FATAR detected the logical End-of-Tape (EOT) on the output tape, and EOTBLOCKS= was specified or defaulted.

Action: Since an image copy must put all the input data on a single output volume, FATAR will attempt to write "nnn" additional blocks beyond logical EOT.

FATS078 **EOT ON TAPEOUT, FATAR WROTE nnn MORE BLOCKS BUT COPY NOT COMPLETE**

Reason: During an image copy using LABELS=NO, FATAR wrote extra blocks beyond logical EOT, but the allowable block count was reached before all input data was copied.

Action: Since an image copy must put all the input data on a single output volume, this copy is not complete. If "nnn" is less than 999, you can attempt to complete it by increasing the EOTBLOCKS= value, or using a longer output tape.

FATS079 **PHYSICAL EOT ON TAPEOUT AFTER nnn EXTRA BLOCKS WRITTEN, COPY INCOMPLETE**

Reason: During an image copy using LABELS=IMAGE, FATAR wrote extra blocks beyond logical EOT, but physical EOT was detected before all input data was copied.

Action: Since an image copy must put all the input data on a single output volume, this copy is not complete. You can attempt to complete it by using a longer output tape.

FATS080 **UNABLE TO LOAD CHARTAB=table PRINT TRANSLATE TABLE, DEFAULT "FATCHRTB" WILL BE USED.**

Reason: CHARTAB= was specified with the name of a load module containing an alternate print table, but FATAR was unable to successfully LOAD that module.

Action: Check the job log for IBM messages indicating why the module could not be loaded. For this run the standard FATAR translate table was used.

FATS081 **TAPE LABELS IN ISO/ANSI FORMAT - TRANSLATED TO EBCDIC FOR PRINTING ONLY**

Reason: ANSI labels were detected on the input tape. They are translated to EBCDIC for printing purposes only.

FATS082 **READ BUFFER NOT AVAILABLE - USE DUMP=YES OPTION**

Reason: This is an internal error. Contact Innovation for assistance.

FATS083 ****WARNING** tapedd REWOUND TO LOAD POINT FOR IMAGE COPY**

Reason: An image copy was requested (LABELS=IMAGE for FATAR, IMAGE for FATSCOPY), but **tapedd** (TAPEIN or TAPEOUT) was not positioned at the load point. For both FATAR and FATSCOPY, this could be because a value greater than 1 was specified in the LABEL= parameter on the DD statement in the JCL. For FATSCOPY only, LABEL=(,SL) was specified (or defaulted) on the TAPEOUT DD.

Action: The tape was rewound to the load point (beginning of the tape) to copy the entire tape, including the volume and header labels for the first file.

FATS084 ****WARNING** MORE THAN ONE INPUT VOLUME FOUND DURING IMAGE COPY. COPY STOPPED AT END OF FIRST VOLUME**

Reason: An image copy was requested (LABELS=IMAGE) but the TAPEIN JCL specified more than one volume serial. Image copy can process only one volume at a time.

Action: The additional volume serials were ignored. If they must also be copied, you must submit separate LABELS=IMAGE steps for each one.

FATS085 ****WARNING** NUMFILES SET TO 0 FOR IMAGE COPY**

Reason: An image copy was requested (LABELS=IMAGE) but NUMFILES= (NF=) was specified with a value other than 0.

Action: Since image copy must copy the entire tape, NUMFILES=0 was forced.

FATS086 ****WARNING** NO TAPEOUT DD FOUND WHEN LABELS=IMAGE SPECIFIED**

Reason: An image copy was requested (LABELS=IMAGE) but no TAPEOUT DD was provided to specify the output tape.

Action: Processing will continue, allowing you to "dry run" the image copy, but the step must be rerun with a TAPEOUT DD statement to actually copy the tape.

FATS087 ****WARNING** CATALOG OF OUTPUT TAPE FAILED. COMP=comp DSN=dsn**

Reason: Cataloging of the indicated output data set was requested by the CAT=YES or CAT=RECAT parameters, but the attempt to catalog the indicated data set failed. "comp" is the return code from CAMLST CATALOG, which will usually be 0008. For CAT=YES this may indicate that the output data set name is already cataloged. Another common cause is attempting to catalog into the master catalog, usually because you have used a output data set name that is not assigned to a user catalog. FATAR or FATSCOPY will complete processing, but the indicated output data set names will not be cataloged.

Action: You can catalog the output data set name manually without rerunning the job, using the IDCAMS statement DEFINE NONVSAM. If you want to recatalog the output name to the output tape, rerun the step using CAT=RECAT instead of CAT=YES. If the output name is not valid, use the RENAME statement to change the name to one that can be cataloged.

FATS088 ****WARNING** CATALOGING REQUESTED BUT FULL NAME NOT AVAILABLE. DSN=dsn**

Reason: Cataloging of the indicated output data set was requested by the CAT=YES or CAT=RECAT parameters, but was not attempted because FATAR does not know the full 44-character name of the data set. IBM records only the last 17 characters of the original data set name in the tape labels, which is not valid for cataloging. FATAR can attempt cataloging if:

- the FATAR tape management interface is enabled and has provided the full data set name from tape management records
- the name in the labels is 16 characters or less
- you have provided the output data set name on a RENAME statement with the NEWN= parameter.

FATAR will complete processing but the indicate output data set names will not be cataloged.

Action: You can rerun the FATAR step providing RENAME statements with NEWN= to provide the output data set name. If the tape is recorded in your tape management system but you have not enabled the FATAR tape management interface, you can do so (See Section 90.3).

FATS089 ****WARNING** NEWI= SPECIFIED BUT FULL NAME NOT AVAILABLE. NEWI=mask**

Reason: A RENAME statement with the NEWI= operand applies to this file, but the full data set name of the input data set was not available. The full name is available if the original name is 16 characters long or less, if the full name was provided on the TAPEIN DD statement, or if you have enabled the FATAR tape management interface.

Action: The output file will be created with the truncated name derived from the input tape labels. You may want to rerun the copy specifying a full output data set name with NEWN= instead of NEWI=.

FATS090 ****WARNING** NEWI= RESULTED IN INVALID OUTPUT DATASET NAME. NEWI=mask DSN=dsn VOL=volser**

Reason: A RENAME statement with the NEWI= operand applies to this file, but when the mask was applied to the input data set name, it resulted in an output data set name that does not meet IBM standards. The volume and dsn are displayed only by FATSCOPY.

Action: The output data set will be created with the input data set name, ignoring the RENAME statement. You may want to rerun the copy specifying a NEWI= which results in a valid name, or specify a NEWN= to provide a complete output name.

FATS091 ****WARNING** NEXTVOL=TMS SPECIFIED BUT TMS INFORMATION NOT AVAILABLE.**

Reason: NEXTVOL=TMS was specified, requesting that FATAR get information about the additional input tape volumes related to the first input tape from the FATAR tape management interface, but no tape management information was available for this tape. Either the tape management interface is not enabled (See Section 90.3) or the input tape is not recorded in the tape management database.

Action: If additional volume serials were specified on the TAPEIN DD statement, they will be mounted. Otherwise, FATAR will terminate normally at the end of the input tape.

FATS092 **OUTPUT DATASET RENAMED TO DSN=dsname**

Reason: A RENAME statement was supplied that applies to this file. The actual name of the output data set is displayed.

FATS093 **OUTPUT DATASET RENAMED AND CATALOGED. FILE=ffff,UNIT=xxxxxxxx,DSN=dsname**

Reason: A RENAME statement was supplied that applies to this tape file and CAT= was also specified. The actual name of the output data set is displayed; this data set was cataloged with the file number and device type (in hex) shown.

FATS094 ****WARNING** CATALOGING REQUESTED BUT NEW NAME WAS NOT SPECIFIED.**

Reason: CAT= was specified, requesting that this output data set be cataloged, but the full name of the input data set was not available and no RENAME NEWN= was provided with an output data set name. The full name is available if the original name is 16 characters long or less, if the full name was provided on the TAPEIN DD statement, or if you have enabled the FATAR tape management interface.

Action: The output file will be created with the truncated name derived from the input tape labels but this name was not cataloged.

FATS095 **OUTPUT DATASET CATALOGED. FILE=ffff,UNIT=xxxxxxxx,DSN=dsname**

Reason: CAT= was specified, requesting that this output data set be cataloged, and the current output data set was cataloged with the file number and device type (in hex) shown.

FATS096 **END OF DATA MARK ENCOUNTERED. BLOCK ID=xxxxxxxx. VOL=volser DSN=dsn**

Reason: FATAR or FATSCOPY encountered an End-of-Data (EOD) hardware mark on an input tape. An EOD mark is written on every cartridge (except 3480-format) after the trailer labels of the last file on the tape, so this indicates that your FATAR options requested FATAR to read beyond that last file. "xxxxxxxx" is the hardware block ID of the EOD mark. The volume and dsn are displayed only by FATSCOPY.

Action: FATAR will attempt to position past the EOD mark and continue. However, the results may vary depending on the type of tape and the vendor of the hardware; sometimes it may be unable to move past the EOD and other times it may need to skip over some data past the EOD. FATSCOPY will stop processing the current file and proceed to the next data set.

FATS097 **LABELS=ONLY SPECIFIED. DATA FILE WILL BE SKIPPED.**

Reason: LABELS=ONLY was specified on the ANALYZE/COPY statement. FATAR will read and format tape labels, but will use a Forward Space File CCW to rapidly skip over data files. No meaningful statistics about the contents of the data files will be printed.

FATS098 **LABELS=ONLY SPECIFIED. NO TOTALS WILL BE PRODUCED FOR THIS RUN.**

Reason: LABELS=ONLY was specified on the ANALYZE/COPY statement. FATAR will read and format tape labels, but will use a Forward Space File CCW to rapidly skip over data files. FATAR totals are meaningless without information about the data files, so they are not printed.

FATS099 **NEAR RELATIVE DATA LOCATION nnn - error description**

Reason: An error occurred while parsing a control statement. The approximate location of the error was position nnn, counting the first position as 000. The failing statement is normally printed immediately above.

Action: Correct the error described in “error description” and re-submit the job.

FATS100 **(INSTRUCTION LINE)**

Reason: If PARM=I is present on the FATS EXEC JCL statement, this message will be printed repeatedly on SYSPRINT with the internal FATS instructions.

FATS101 ****PHYSICAL VOLUME=volser CONTAINED NO LOGICAL VOLUMES.**

Reason: PHYSVOL= was specified on a SELECT statement, but a check of the Virtual Tape Server found no logical volumes associated with the physical volume.

Action: FATSCOPY will set a return code '8' to indicate that an error occurred. FATSCOPY will continue to parsecontrol statements and will process any volumes that matched other SELECT statements.

FATS102 **CONTROL CARD ERROR -- REASON=x**

Reason: A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS102 message will be printed for each input statement; multiple errors on a statement will not be diagnosed. "x" is a letter or number indicating the type of error detected:

1 -- An option parameter is not followed by a blank or comma.

2 -- Operation keyword is invalid, or the tape number in parentheses is not 1 through 9.

3 -- Tape number has been previously used on another operation statement.

4 -- An option parameter is invalid.

5 -- Error in the MODE= parameter. Value is not 2 hexadecimal digits or is not one of the valid values.

6 -- Error in the RETRY= parameter. Must be RETRY=h or RETRY=(l,h). "h" must be 1 to 99. "l" must be 0 to "h".

7 -- Error in the numeric value of a keyword parameter. Invalid numeric digits.

8 -- Error in the BPI= parameter, BPI=0.

9 -- Error in the LABEL= parameter, invalid characters in the label value.

A -- Expected continuation statement not received (previous statement ended in a comma).

B -- No ending apostrophe in OWNERID= parameter.

C -- Error in the THRESHOLD= parameter, greater than 32767.

D -- Label operation statement missing LABEL= or VOL= parameter.

E -- Error in the BLKSIZE= parameter; it must be less than 65536 and more than twice the tape density (BPI), except for cartridges.

F -- Error in VOL= parameter. Volume serial longer than 6 characters.

G -- Error in VOL= parameter. More than 682 volume serials specified.

H -- VOL= parameter specified on a default statement.

I -- VOLINCR= parameter specified without VOL= parameter.

J -- Blank control statement.

K -- Error in VOLINCR= parameter. The last volume serial in the VOL= parameter could not be incremented due to insufficient trailing numeric digits.

L -- Error in ENDAFTER= parameter.

M -- Invalid numeric value for keyword. Value was zero or greater than 32767.

N -- An option parameter was specified that is invalid for the control statement on which it was found. For example, MODIFY=YES was specified on a read statement, but it is only valid on a DEFAULT statement.

Action: FATS will abend with a U0502 abend code after all control statements have been scanned.

FATS103 ****TAPESUMM DD STATEMENT MISSING. ADD TAPESUMM DD TO JCL AND RERUN JOB.**

Reason: FATSCOPY requires a TAPESUMM DD.

Action: FATSCOPY will abend with a U0888. Add a TAPESUMM DD to your JCL and rerun the job.

FATS104 **ONE OR MORE ERRORS DETECTED -- FATS TERMINATED**

Reason: Message FATS102 was issued for one or more control statement errors.
Action: FATS will abend with a U0502 abend code. Correct the errors and re-execute.

FATS105 **NO INPUT CARDS -- FATS TERMINATED**

Reason: No control statements other than DEFAULT and comments were present in the input. At least one operation statement must be present.
Action: FATS will abend with a U0502 abend code.

FATS106 **DD=SYSPRINx MISSING -- REQUIRED FOR ANALYZE**

Reason: An ANALYZE(x) control statement was present but no corresponding SYSPRINx DD statement was found. FATAR cannot write its messages to SYSPRINT when invoked from FATS.
Action: FATS will abend with a U0502 abend code.

FATS107 **A line of the detail report indicating label status**

FATS108 **NO SUMMPRT DD PROVIDED-SUMMARY REPORT WILL GO TO SYSPRINT**

Reason: The SUMMPRT DD statement is not present.
Action: FATS will write summary report data to SYSPRINx if present for TAPEx or to SYSPRINT

FATS109 **FATS BYPASSED DD=TAPEx REASON=MISSING DD STATEMENT DEVICE NOT A TAPE**

Reason: FATS was unable to process an operation statement because the associated DD statement TAPEx was not provided or did not allocate a tape device.
Action: Other operation statements (if present) will be processed. Check the TAPEx DD statement.

FATS110 **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

Reason: FATS has read the control statement printed as part of this message.
Action: FATS will parse the control statement.

FATS111 **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

Reason: FATS has parsed all of the control statements and merged the information for each with the defaults. This table represents the processing to be done for each of the 1 to 9 functions specified.
Action: FATS will perform the requested functions.

FATS112 **FATS MUST BE IN AN AUTHORIZED LIBRARY. PROGRAM TERMINATED.**

Reason: FATS must be executed as an APF authorized program. Either the program library containing FATS was not marked as an APF authorized library on your z/OS system, or your STEPLIB or JOBLIB DD statement has nonauthorized libraries concatenated to the FATS program library, or FATS has been relinked as nonauthorized (AC=0).
Action: FATS will terminate with a U0047 ABEND.

FATS113 **PROGRAM TERMINATED. USER NOT AUTHORIZED TO PERFORM FUNCTION=func**

Reason: Your installation has protected certain functions and options of FATS and FATAR through your security system. You are not authorized to use the function or option "func".
Action: The program will terminate with a U0913 abend. If you require the failing function or option, contact your security administrator. See Sections 11.8 (FATS) and 21.6 (FATAR) for information on the security options.

FATS114 PROGRAM TERMINATED. USER NOT AUTHORIZED TO READ DSN=dsname

Reason: Your security system has indicated that you are not authorized to read a file on the input tape with data set name "dsname".

Action: The program will terminate with a U0913 abend. If you require access to this data set, contact your security administrator. See Section 21.6 for information on the FATAR data set security.

FATS115 UNLABELED TAPE USED AS INPUT WHEN LABELS=ONLY WAS SPECIFIED. RUN TERMINATED.

Reason: LABELS=ONLY is supported only on a labeled tape.

Action: The program will terminate.

FATS116 UNABLE TO ACCESS DATA PAST END-OF-DATA MARK. RUN TERMINATED.

Reason: All cartridge tapes (except 3480) put a hardware EOD (End-of-Data) mark after the last block or tape mark written to the tape. This message indicates that FATAR was unable to position past the EOD mark. This can occur when:

- LABELS=EOD was specified, to recover data past the EOD
- LABELS=NO and NUMFILES=n was used to request processing past the EOD.

Action: The program will terminate with a U0888 abend.

FATS117 ANALYSIS TERMINATED AT END-OF-DATA MARK

Reason: On a cartridge tape (except for 3480), FATAR encountered a EOD (End-of-Data) mark that follows a tape mark.

Action: Processing of the input tape is completed.

FATS117 COPY TERMINATED AT END-OF-DATA MARK DSN=dsname, VOL=volser

Reason: FATSCOPY encountered an end-of-data mark when copying a data set. Output data set will not be cataloged or updated with tape management information from the input data set. If TMSINPUT=SCRATCH was specified, the input will not be expired.

Action: FATSCOPY will continue with the next data set.

FATS118 UNABLE TO LOAD name ASCII TRANSLATE TABLE. DEFAULT IBM TABLE WILL BE USED

Reason: The ASCII=name operand was specified, but a load module called "name" could not be successfully loaded, so the default IBM ASCII translate table was used.

Action: Make sure that the load module name of the ASCII translate table was properly specified, and that it is in a load module library accessible by this FATAR step.

FATS120 **DSN=dsname ON VOLUME=volser WAS SELECTED FOR COPYING.

Reason: FATSCOPY selected the indicated data set on the indicated tape volume from the system catalog for copying.

FATS121 **NO DATA SETS WERE SELECTED FOR COPYING. JOB TERMINATED.

Reason: The SELECT/EXCLUDE statements specified for FATSCOPY did not result in any data sets being selected.

Action: If the statements were specified incorrectly, correct and re-submit. If data sets should have been selected, check the statements carefully for errors; if necessary, contact Innovation for assistance.

When the FATS121 message is issued, the FATSCOPY job normally terminates with a U0300 abend and produces a dump. To terminate with a return code of 12 instead, and no dump, use the RETCODE parameter. To terminate with a a return code of 0 instead, and no dump, use SELTERR=NO.

FATS122 ****FATSCOPY SIMULATE SUCCESSFULLY COMPLETED. nnnn DATA SETS WERE SELECTED FOR COPYING.**

Reason: FATSCOPY was executed with a SIM (SIMULATE) statement. If it had been executed with a COPY statement, "nnnn" data sets would have been copied. The selected data sets are listed in the SELRPT and COPYRPT reports.

FATS123 ****LOAD OF RMM API FAILED. DSN=dsname VOL=volser WILL NOT BE EXPIRED.**

Reason: Your tape management system is apparently DFSMSrmm (RMM) but the RMM Application Program Interface (API) could not be used by FATSCOPY to expire the data set.

Action: Contact Innovation for assistance.

FATS124 ****SELRPT DD STATEMENT MISSING. ADD SELRPT DD TO JCL AND RERUN JOB**

Reason: You omitted the required SELRPT DD statement in a FATSCOPY step. The step will end with a U0888 abend.

Action: Add the SELRPT DD to your step; it is usually a SYSOUT=* data set.

FATS125 ****COPYRPT DD STATEMENT MISSING. ADD COPYRPT DD TO JCL AND RERUN JOB**

Reason: You omitted the required COPYRPT DD statement in a FATSCOPY step. The step will end with U0888.

Action: Add the COPYRPT DD to your step; it is usually a SYSOUT=* data set.

FATS126 ****MAXIMUM NUMBER OF INPUT TAPE FILES EXCEEDED. SELECT STATEMENT NOT PROCESSED.**

Reason: The preceding SELECT statement was ignored because the maximum number of files to be copied in this FATSCOPY step been exceeded. For CATDSN, the limit is specified by MAXTOTFILE=. For ALLDSN, the limit is 65,535.

Action: Include the SELECT statement(s) in another FATSCOPY step to copy those data sets, or specify a larger value for MAXTOTFILE=.

FATS127 ****NO TMS INFORMATION FOUND. TMS INFO WILL NOT BE COPIED DSN=dsname VOL=volser**

Reason: The indicated data set was selected from your system catalogs by FATSCOPY, but no tape management information was found for the data set.

Action: The data set will be copied but no TMS information will be updated for this data set.

FATS128 ****RMM API LOAD FAILED. DATA FOR DSN=dsname VOL=volser WILL NOT BE COPIED**

Reason: Your tape management system is apparently DFSMSrmm (RMM) but the RMM Application Program Interface (API) could not be used by FATSCOPY.

Action: Contact Innovation for assistance.

FATS129 ****DYNAMIC ALLOCATION FAILED FOR VOL=volser DEVTYPE=devtype. RC=rrrr, REAS=cccc0000**

Reason: FATSCOPY issued a dynamic allocation for an input tape file to copy on the volume serial and tape device type shown, but the allocation failed with the indicated return code and reason code.

Action: The tape file will be bypassed; serious errors may cause the step to abend. For z/OS, these codes are documented in the IBM manual *Authorized Assembler Services Guide*. They can also be found in Appendix A of the ISPF online HELP. Frequently encountered "cccc" values include:

0210 - requested data set not available (e.g., another job had the tape file allocated with DISP=OLD).
021C - invalid unit name used (e.g., FATSCOPY is trying to allocate a 3480 on a system without 3480s).
0220 - requested volume not available (e.g., another job is using a tape volume required by FATSCOPY).
039C - requested device type and volume are incompatible. This may occur if the data set is cataloged to a device type that does not match the volume it is located on. Specifying UNITIN= on the COPY statement may resolve this problem.

FATS130 ****DYNAMIC DEALLOCATION FAILED FOR VOL=volser DEVTYPE=devtype. RC=rrrr, REAS=cccc**

Reason: FATSCOPY issued a dynamic de-allocation for an input tape file to copy on the volume serial and tape device type shown, but the de-allocation failed with the indicated return code and reason code.

Action: FATSCOPY will terminate. For OS/390, these codes are documented in the IBM manual *Authorized Assembler Services Guide*. They can also be found in Appendix A of the ISPF online HELP.

FATS131 ****EXPIRATION DATE FOR INPUT VOLUME=volser HAS BEEN CHANGED: date**

Reason: The user specified TMSINPUT=SCRATCH; the expiration date for the volume has been changed to the indicated date so that the volume will expire on that date and be scratched when tape management runs.

FATS132 ****TAPE MGMT INFO FOR DSN=dsname ON VOL=output WAS COPIED FROM VOL=input**

Reason: FATSCOPY has copied the appropriate information, such as expiration date, creating job, etc. from the tape management record for the input volume to the tape management record for the output volume.

FATS133 ****EXPIRATION DATE FOR DSN=dsname,VOL=volume CHANGED TO date**

Reason: FATSCOPY has changed the expiration date for the output data set on the indicated output volume to the date specified. This is the latest date for any data set on the volume. This message is issued for CA 1 and TLMS, and also for RMM if RMM data set expiration support is available (message FATS165 is issued if it is not available).

FATS134 ****UNABLE TO CHANGE EXPIRATION DATE FOR OUTPUT VOLUME=volser RC=return code, REASON=reason code**

Reason: FATSCOPY's attempt to update the tape management expiration date for the output volume failed.

FATS135 ****UNABLE TO EXPIRE INPUT type=name RC=return code,REASON=reason code**

Reason: FATSCOPY tried to expire the input volume (type="VOLUME") or data set (type="DATASET") but was unable to do so.

FATS135 ****UNABLE TO RESET KEEP DATE FOR INPUT CONTROL DATA SET, VOLUME=volser**

Reason: TMSINPUT=SCRATCH was specified, but FATSCOPY was unable to change the expiration date for the TLMS control data set on the input volume.

FATS136 ****UNABLE TO DELETE RMM DATASET INFO FOR OUTPUT DSN=dsname VOLUME=volser**

Reason: While copying a data set, FATSCOPY reached the end of tape on the OUTPUT. At the time it reached end of tape, the number of bytes written to TAPEOUT was less than the value specified in the LASTFILE= parameter. FATSCOPY attempted to delete the tape management information for the output data set, but failed to do so.

FATS137 ****NO TMS INFO FOR DSN=dsname. UNABLE TO EXPIRE INPUT VOL=volser**

Reason: FATSCOPY copied a cataloged data set that had no corresponding tape management information. FATSCOPY will be unable to update the tape management information for the output data set.

FATS138 ****FATSCOPY UNABLE TO CHANGE EXPIRATION FOR VOLUME=input volser. ALL DATA SETS ON xxxxxx NOT SELECTED FOR COPYING**

Reason: TMSINPUT=SCRATCH was specified, but some of the files on the volume or tape set were not copied. ”xxxxxx” will be “INPUT” if the identified volume was a multi-file tape or “MVOL SET” if the volume is part of a multi-volume, multi-file tape set.

Action: FATSCOPY updated the expiration dates of the copied files, but could not update the expiration date of the volume(s) because some unexpired files still remained on those tapes. A return code of 4 will be issued.

FATS139 ****RMM UPDATE FAILED. VOLUME=volser DSN=dsname RC=return code,REAS=reason code.**

Reason: TMSDATA=COPY was specified, but FATSCOPY’s attempt to update the tape management information for the volume failed. The displayed return and reason codes are the hex values returned by RMM.

FATS140 ****COPY FUNCTION SPECIFIED BUT NO TAPEOUT DD FOUND. JOB TERMINATED.**

Reason: The COPY statement requires a TAPEOUT DD statement.

FATS141 ****TMS NOT ACTIVE. TMS DATA WILL NOT BE COPIED. DSN=dsname,VOL=volser**

Reason: The tape management system expected was not active, so FATSCOPY will not attempt to copy tape management data relating to the data sets it is copying.

FATS142 ****ERROR OCCURRED READING TMC FOR xxxxxx VOL=volser. TMS DATA WILL NOT BE COPIED.**

Reason: An error occurred while reading tape management data for the input tape (xxxxxx=INPUT) or the output tape (xxxxxx=OUTPUT). The tape management data for the file being copied will not be updated on the output tape's records.

FATS142 ****ERROR OCCURRED READING VMF FOR OUTPUT VOL=volser. UNABLE TO OBTAIN BLOCK INFO.**

Reason: An error occurred while reading TLMS block count information for the output volume. Block and byte count totals for the output tape will not be correct in the summary reports.

FATS143 ****ERROR OCCURRED UPDATING TMC FOR DSN=dsname VOL=volser**

Reason: An error occurred while updating tape management data for the output data set. The tape management data for the file being copied will not be updated on the output tape's records.

FATS143 ****ERROR OCCURRED UPDATING VMF FOR DSN=dsname VOL=volser REASON=xxxxxxxx**

Reason: An error occurred while updating the TLMS Volume Master File for the output data set. The tape management data for the file being copied will not be updated on the output tape's records. xxxxxxxx is the TLMS error message code.

FATS144 ****ERROR OCCURRED DEQUEUING TMC FOR DSN=dsname VOL=volser**

Reason: An error occurred while updating tape management data for the output data set. The tape management data for the file being copied will not be updated on the output tape's records.

FATS145 ****DSN=dsname ON VOL=volser BYPASSED - MISSING VOLS OF MULTI-VOL DSN.**

Reason: When processing a data set that had met all the selection criteria, FATSCOPY determined that some volumes of a multi-volume set had not met the selection criteria. FATSCOPY will not copy the data set.

FATS146 ****UNSUPPORTED TAPE MGMT SYSTEM. UNABLE TO UPDATE TAPE MANAGEMENT INFORMATION**

Reason: TMSDATA=COPY or TMSINPUT=SCRATCH was specified, but FATSCOPY didn't find CA 1, TLMS, or RMM, the only tape management systems currently supported.

FATS147 ****TAPE MOTION I/O ERROR**

Reason: An I/O error occurred while FATSCOPY was reading an input tape. FATSCOPY was unable to reposition the tape to retry the I/O that had the error.

Action: FATSCOPY will terminate with a U0888 abend.

FATS148 ****TMC OPEN FAILED. TMS DATA CAN NOT BE COPIED. DSN=dsname,VOL=volser**

Reason: An error occurred when FATSCOPY tried to open the TMC to obtain tape management information for the input volume to update tape management information for the output volume. The TMS information will not be copied to the output.

FATS148 ****TMC OPEN FAILED. DATA SET WILL NOT BE EXPIRED. DSN=dsname,VOL=volser**

Reason: An error occurred when FATSCOPY tried to open the TMC to expire the input data set.

FATS148 ****VMF OPEN FAILED. UNABLE TO UPDATE TLMS INFO FOR DSN=dsname,VOL=volser**

Reason: An error occurred when FATSCOPY tried to open the Volume Master file to expire an input data set or update the TLMS information for an output data set.

FATS148 ****VMF OPEN FAILED. UNABLE TO OBTAIN BLOCK INFO FOR REPORT.**

Reason: An error occurred when FATSCOPY tried to open the TLMS Volume Master file to read block or byte count information for the output file summary reports.

FATS149 ****TMS ERROR CODES: RC=return code,REAS=reason code,SEC RC=security return code, SEC REAS=security reason code**

Reason: Issued after an error reading, updating, or dequeuing the TMC occurs. Contains CA 1 abend codes, reason codes, and security return and reason codes.

FATS150 ****TMS NOT ACTIVE. INPUT TAPE WILL NOT BE EXPIRED. VOL=volser**

Reason: TMSINPUT=SCRATCH was specified, but TMS was not active.

FATS150 ****TMS NOT ACTIVE. INPUT DSN=dsname, VOL=volser WILL NOT BE EXPIRED.**

Reason: TMSINPUT=SCRATCH was specified, but TMS was not active.

FATS151 ****ERROR OCCURRED xxxxxxx TMC FOR INPUT VOL=volser. TAPE WILL NOT BE EXPIRED.**

Reason: An error occurred when FATSCOPY tried to obtain the expiration date for the input volume prior to changing the expiration date to the current date (xxxxxxx = READING) or when FATSCOPY tried to update TMS with the new expiration date (xxxxxxx = WRITING). The expiration date will not be changed.

FATS151 ****ERROR UPDATING TAPE MGT. DSN=dsname VOL=volser WILL NOT BE EXPIRED.**

Reason: TMSINPUT=SCRATCH was specified, but an error occurred when FATSCOPY tried to to update TMS with the new expiration date. The expiration date will not be changed for this data set.

FATS152 ****EOT ON TAPEOUT BEFORE LASTFILE VALUE REACHED. DSN=outdsn,VOL=outvol**

Reason: The physical end-of-tape was reached before FATSCOPY copied the number of bytes specified in the LASTFILE= keyword. FATSCOPY will call for a new output tape, determine which tape the input file started on, and begin copying the input file from the beginning. If tape management is active, FATSCOPY will expire the partially copied data set on the output tape.

FATS153 ****PARTIALLY COPIED DSN=outdsn ON OUTPUT VOL=outvol WAS EXPIRED**

Reason: FATSCOPY successfully expired the partially copied data set on the output tape in tape management.

FATS154 ****NO ACTIVE TAPE MANAGEMENT. DSN=outdsn, VOL=outvol WILL NOT BE EXPIRED.**

Reason: FATSCOPY attempted to expire a data set in tape management that had been partially copied, but tape management system was not active.

FATS155 ****UNABLE TO FIT DSN=outdsn ON ONE VOLUME. LASTFILE= VALUE WILL BE IGNORED.**

Reason: After calling for a new output volume for a data set that wouldn't fit on the output volume, FATSCOPY again arrived at the end of the output tape without exceeding the LASTFILE= amount. FATSCOPY will call for a new output without starting to copy the input from the beginning.

FATS156 ****EXPIRATION DATE FOR DSN=ipdsn, VOL=ipvol WAS SET TO DATE=expirationdate**

Reason: After copying a data set, FATSCOPY expired the input data set because TMSINPUT=SCRATCH was specified. The indicated data set and volume were set to the indicated expiration date.

FATS157 ****EXPIRATION DATE FOR INPUT VOLUME=volser WAS RESET TO DATE=date**

Reason: When FATSCOPY has copied the first data set (if the tape management system is CA 1) or the control data set (if the tape management system is TLMS) on a volume or multivolume set containing multiple files, FATSCOPY will reset the expiration date for the volume to the highest expiration date of any unexpired data set on the volume / multivolume set. If it copies any other files after copying the first file, FATSCOPY will again reset the volume to the highest expiration date of any remaining data sets. If all the data sets on the volume are successfully copied, FATSCOPY will expire the volume at that time.

FATS158 ****ERROR READING TMC. DSN=dsname, VOL=volser WILL NOT BE EXPIRED.**

Reason: OPEN of the CA 1 TMC for update by FATSCOPY failed and it will not be able to expire the data set. This message will be followed by a FATS149 containing the error codes returned by TMS for the failed OPEN. This may be due to security checks in CA 1; check the CA 1 documentation for information on security protection of the TMC and the YSVCUNCD resource.

FATS159 ****NO DATASETS COPIED DURING FATSCOPY RUN. CHECK LISTING FOR ERROR MESSAGES.**

Reason: FATSCOPY ended with no data sets being copied, even though data sets were selected for copying. The SYSPRINT or ERRPRRPT report will contain error messages to show why no data sets were copied.

FATS160 ****OPEN FAILED FOR DSN=dsn,VOL=volser,ABEND=abend code**

Reason: FATSCOPY tried to open an input data set, but the open failed with the ABEND code shown. The joblog may contain IBM messages providing more detail on the error.

Action: The input data set will be skipped and FATSCOPY will continue to copy other data sets, if any are left.

FATS161 ****EXPIRATION FOR DSN=dsn,VOL=volser WILL BE RESET BY RMM HOUSEKEEPING.**

Reason: While attempting to update the RMM expiration date for a data set it just copied, FATSCOPY found that the data set's expiration was controlled by a RMM Vital Record Specification. The next time RMM housekeeping runs, RMM will set the expiration for the output data set according to the VRS.

FATS162 ****MAXIMUM NUMBER OF FILES SELECTED FOR COPYING. CATALOG SEARCH STOPPED.**

Reason: While searching the catalog for data sets to copy, FATSCOPY exceeded the maximum number of data sets specified by the MAXTOTFILE= operand (default: 1000).

Action: FATSCOPY will stop selecting data sets and will process the data sets it has selected. Rerun FATSCOPY to select the remaining data sets, and increase MAXTOTFILE= if needed.

FATS163 ****CATALOGING BYPASSED FOR UNCATALOGED INPUT DATA SET, DSN=dsname**

Reason: CAT=ONLY was specified (or defaulted in FATSCOPY) and an input data set was not cataloged. This could mean that the input data set was not in the catalog at all, or that it was cataloged but it pointed to a volume or file number different from the input data set.

Action: The output file will not be cataloged. A return code based on the value specified or defaulted for NOTCATRC= (0, 4, or 8) will be issued.

FATS164 ****JOB TERMINATED DUE TO RMM READ OR UPDATE FAILURE.**

Reason: An update or read of RMM tape management information occurred, and RMMFAIL=STOP had been specified.

FATS165 ****LAST USED INFO FOR VOLUME=oooooo COPIED FROM VOLUME=iiiiii**

Reason: TMSDATA=COPY was specified, so FATSCOPY copied the tape management information about the last use of the input tape "iiiiii" to the output tape "oooooo"

FATS166 ****EXPIRATION DATE FOR OUTPUT VOLUME=outvol HAS BEEN SET TO expirationdate|VITAL RECORD**

Reason: Documents the expiration date that has been assigned to the indicated output volume. If the tape management system is RMM and one of the input data sets had a VITAL RECORD expiration, the message will show VITAL RECORD. If the tape management system is TLMS, this message is issued when the expiration date of the control data set (by default, data set #1) has been set on the output volume.

FATS167 ****SELECT ALLDSN SPECIFIED BUT NO TAPE MANAGEMENT INFO FOR VOLUME=volser. VOLUME BYPASSED.**

Reason: A SELECT ALLDSN statement was used to select an input tape, but no tape management information was found for the indicated volume.

Action: The volume is bypassed. If there are no more SELECT statements or they all fail the same way, the step will end with a U0300 ABEND.

FATS168 ****ALL DATA SETS ON VOL=input volser COPIED. VOLUME RELEASED.**

Reason: RMM was the active tape management system, the input volume had a RMM Vital Record Retention and all data sets on the indicated input volume have been copied. TMSINPUT=SCRATCH was specified.

Action: RMM is notified to release the input tape.

FATS169 ****RELEASE FOR VOL=input volser FAILED. RC=return code, REAS=reason code**

Reason: RMM was the active tape management system, the input volume had a RMM Vital Record Retention and all data sets on the indicated input volume have been copied. TMSINPUT=SCRATCH was specified, but an error occurred when RMM was notified to release the tape.

Action: The input tape is not released. You may need to release it manually.

FATS170 ****LASTAPE DATA SET CATALOGED. VOL=volser,DSN=dsname**

Reason: LASTAPE= was specified. At the end of the FATSCOPY step, the data set name specified was created as an empty file and cataloged to the volume shown.

FATS170 ****ARCHIVE CONTROL FILE BACKUP CATALOGED. VOL=volser,DSN=dsname**

Reason: ARCBACKUP=DSF was specified. At the end of the FATSCOPY step, a backup of the Archive Control File was created and cataloged to the volume shown using the data set name specified by the ARCB1DSN= parameter (or the default).

FATS171 ****EXPDTGROUP VALUE SET TO 999 FOR LASTAPE PROCESSING.**

Reason: FATSCOPY normally groups data sets by expiration date. When LASTAPE= is specified, EXPDTGROUP will be set to 999.

Action: Data sets will be not be grouped by expiration date.

FATS172 ****reason. UNABLE TO DETERMINE IF LASTAPE VOLUME EXCEEDS THRESHOLD.**

Reason: LASTAPE= was specified with THRESHOLD=, but FATSCOPY was unable to determine if the percentage used of the output tape exceeded the specified threshold. "**reason**" may be:

NO ACTIVE TAPE MANAGEMENT – No tape management system was active.

UNSUPPORTED TAPE MANAGEMENT – THRESHOLD is not supported with the active tape management system.

TMC OPEN FAILED – Open of the CA 1 TMC failed; message FATS149 follows.

Action: No threshold checking is done. FATSCOPY will continue to add files to the tape. A return code of 8 will be set.

FATS173 ****TMS REQUEST FOR LASTAPE VOLUME volser FAILED. THRESHOLD PROCESING BYPASSED.**

Reason: LASTAPE= was specified with THRESHOLD=, but the LASTAPE data set was cataloged to a tape not under CA 1 control.

Action: No threshold checking is done. FATSCOPY will continue to add files to the tape.

FATS174 ****LASTAPE VOLUME volser EXCEEDED THRESHOLD. NEW OUTPUT VOLUME WILL BE CREATED.**

Reason: LASTAPE= was specified with THRESHOLD=, and the current output volume exceeded the threshold percentage in use.

Action: FATSCOPY will mount a fresh scratch tape.

FATS175 ****LOAD OF RMM API FAILED. THRESHOLD PROCESSING BYPASSED.**

Reason: LASTAPE= was specified with THRESHOLD=, and the current output volume was under DFSMSrmm control but the RMM API module could not be loaded.

Action: No threshold checking is done. FATSCOPY will continue to add files to the tape.

FATS176 ****DEVTYPE=COPYCROS SPECIFIED, BUT NO COPYCROS DEVICES MATCHED VIRTUNIT=TYPE.**

Reason: FATSCOPY was unable to locate any CopyCross devices that matched the specified type.

Action: The step will abend with U0502.

FATS177 ****UNEXPECTED LOAD POINT WHILE COPYING DSN=dsname, VOL=volser**

Reason: When copying a data set, FATSCOPY read the data set, but the input volume had been repositioned to the beginning of the volume. The output data set will not be cataloged or be updated with information for the input data set.

Action: A return code of 12 is set for the current data set.

FATS178 ****UNABLE TO OBTAIN STORAGE REQUIRED FOR TAPE BUFFERS. JOB TERMINATED.**

Reason: FATSCOPY was unable to obtain the storage needed for tape buffers. Make sure that you specified REGION=0M on the EXEC statement. Specify BUFNO= to reduce the number of buffers acquired.

Action: The step will abend with U0888.

FATS179 ****UNABLE TO OBTAIN STORAGE REQUIRED FOR REBLOCKING DATA SETS. JOB TERMINATED.**

Reason: FATSCOPY was unable to obtain the storage needed for reblocking buffers. Make sure that you specified REGION=0M on the EXEC statement. Specify BUFNO= to reduce the number of buffers acquired.

Action: The step will abend with U0888.

FATS180 ****UNABLE TO CREATE LASTAPE DSN DUE TO INVALID CHARACTERS IN DSN. JOB TERMINATED.**

Reason: After processing all the data sets selected for copying, FATSCOPY tried to create the LASTAPE file on the output tape but the LASTAPE data set name was invalid.

Action: The step will abend with S0C3. Contact Innovation for assistance.

FATS181 ****UNABLE TO OBTAIN ELIGIBLE DEVICE FOR ALLOCATING TAPEIN USING nnnnnn. RC=rtncode, REASON=rsncode**

Reason: FATSCOPY was unable to find a unit for mounting the input tape using either the UNITIN= value, the VIRTUNIT value, or the UCB address.

Action: If VIRTUNIT was specified, the step will abend with U0502. Otherwise, the tape is bypassed.

FATS182 ****VERSION 5.4/70 OR HIGHER OF FDRABR REQUIRED FOR COPYING ABR ARCHIVE DATA SETS.**

Reason: FATSCOPY supports copying ABR Archive and Application Backup data sets only if version 5.4/70 or above of FDRABR is available. ABRARC=YES was specified, but a supported level of FDRABR could not be loaded from the linklist, STEPLIB, or JOBLIB.

Action: FATSCOPY will set a non-zero return code, and will not select any ABR Archive or Application backups for copying.

FATS183 ****SERIOUS ERROR OCCURRED OBTAINING JFCB EXTENSIONS FOR TAPEOUT.**

Reason: FATSCOPY clears the TAPEOUT JFCB and JFCB extensions after each data set copy, but a pointer to a JFCB extension was invalid.

Action: The step will abend with U0888.

FATS184 ****LOGIC ERROR OCCURRED DURING LASTFILE PROCESSING. JOB TERMINATED.**

Reason: FATSCOPY was unable to determine the correct volume to use to create the LASTAPE data set.

Action: The step will abend with U0888.

FATS185 ****LOGIC ERROR OCCURRED READING TAPE MGMT INFO FOR DSN=dsname, VOL=volser**

Reason: When attempting to update the output volume with information from the input tape, FATSCOPY was unable to find the correct data set entry for either the input or output data set.

Action: A return code 8 will be set for the failing data set but it will be copied. You may need to update the TMS information manually.

FATS186 ****UNABLE TO EXPIRE INPUT DSN=dsname, VOL=volser DUE TO FATSCOPY LOGIC ERROR.**

Reason: A logic error occurred that prevented FATSCOPY from determining if the input data set could be expired.

Action: A return code 8 will be set for the failing data set but it will be copied. You may need to expire the data set manually.

FATS187 ****LOGIC ERROR OCCURRED DURING THRESHOLD PROCESSING. UNABLE TO DETERMINE IF LASTAPE VOLUME EXCEEDS THRESHOLD.**

Reason: A logic error prevented FATSCOPY from determining the percentage of the last output tape currently in use. Data sets will continue to be added to the tape.

Action: A return code 8 will be set.

FATS188 ****NOT ENOUGH STORAGE FOR FATSCOPY PROCESSING TABLE.**

Reason: The internal data set table used by FATSCOPY was not large enough to contain entries for all the selected data sets.

Action: The job will abend with a U0888. Increase the MAXTOTFILE= value to a higher number, or modify your SELECT statement(s) to select fewer data sets.

FATS189 ****RESTART DATA SET CREATED. DSN=dsname'**

Reason: FATSCOPY created or updated an existing data set that can be used as input to a subsequent FATSCOPY job.

Action: Job will complete with a zero condition code.

FATS190 ****DSNTABLE DD REQUIRED FOR CREATING CHECKPOINT DATA SET OR RESTARTING JOB. JOB TERMINATED.**

Reason: The CHECKPT or RESTART keyword was specified, but a DSNTABLE DD with the name of the data set to use to create the checkpoint data set or restart the job wasn't included in the JCL.

Action: The job will be terminated with a return code '12'. Add a DSNTABLE DD to the JCL and resubmit the job.

FATS191 **** COPY STOPPED BY OPERATOR. nnnnn FILES REMAIN TO BE COPIED.**

Reason: The CHECKPT keyword was specified and an operator issued a STOP command to end a FATSCOPY job. nnnnn indicates the number of files that remained to be copied.

Action: The job will end when it finishes copying the current data set. A checkpoint data set will be created so that the job can be restarted at a later time.

FATS192 ****OPEN OF AUDIT DATA SET FAILED. JOB CANCELLED BY OPERATOR.**

Reason: An AUDITRPT was present in the JCL, but FATSCOPY failed to open the data set, and the operator replied 'NO' to the FATSW16 message issued to the console. .

Action: The job will abend with a U0888.

FATS193 ****UNRECOVERABLE READ ERROR. DSN=dsname,VOL=volser.**

Reason: A read error occurred on an input tape that FATSCOPY was unable to recover from.

Action: FATSCOPY will issue a U0888 for the data set it was copying when the I/O error occurred and continue copying any remaining data sets.

FATS194 ****HSM NOT ACTIVE. UNABLE TO RECYCLE VOLUME=volser**

Reason: HSMML2=YES was specified, but HSM was not active when FATSCOPY issued a request to HSM to recycle a Migration Level 2 data set.

Action: FATSCOPY will set a return code '12' for the current data set and continue with the next data set to be copied.

FATS195 ****TAPE MANAGEMENT INFORMATION FOR OUTPUT VOLUME=volser WAS SET BY HSM RECYCLE.**

Reason: An HSM Migration Level 2 volume was successfully recycled as the result of a recycle request issued by FATSCOPY. All the tape management information for the output volume was set by HSM.

Action: A return code '0' will be set for the data set.

FATS196 ****EXPIRATION DATE FOR OUTPUT VOLUME=volser WAS SET BY HSM RECYCLE.**

Reason: An HSM Migration Level 2 volume was successfully recycled as the result of a recycle request issued by FATSCOPY. HSM has set the appropriate expiration date for the output volume.

Action: A return code '0' will be set for the data set.

FATS197 ****UNABLE TO OBTAIN STORAGE REQUIRED FOR ARCHIVE TABLE. ABR ARCHIVE TAPES WILL NOT BE COPIED.**

Reason: FATSCOPY was unable to obtain the storage needed to perform updates to the Archive Control File.

Action: Any ABR Archive or Application Backup files that were selected will not be copied. A return code '8' will be set and any Archive or Application Backup data sets will not be selected. Increase the REGION size, or code REGION=0M on the EXEC statement.

FATS197 ****UNABLE TO OBTAIN STORAGE REQUIRED FOR ARCHIVE TABLE. JOB TERMINATED.**

Reason: A FATSCOPY RESTART job where ABR Archive or Application Backup files had been selected in the DSNTABLE data set was unable to obtain the storage needed to perform updates to the Archive Control File.

Action: The job is terminated with a U0888 abend. Increase the REGION size, or code REGION=0M on the EXEC statement.

FATS198 ****OUTPUT CREATION DATE RESET AFTER COPY. DSN=dsname,VOL=volser**

Reason: When FATSCOPY is copying a data set under TMS Catalog Control that requires more than 1 output volume, FATSCOPY temporarily sets the creation date for the output to the current date to prevent the output from being expired prematurely by CA 1 housekeeping before FATSCOPY catalogs it. This message is issued to indicate that FATSCOPY has successfully cataloged the output data set and copied the creation date from the input data to the output data set.

FATS199 ****RMM UPDATE FAILED BECAUSE RMM HOUSEKEEPING WAS ACTIVE. FATSCOPY JOB TERMINATED.**

Reason: An update to RMM failed because RMM housekeeping was running; the job was terminated based on the value specified for the RMMFAIL= parameter.

Action: Job will be terminated immediately with a non-zero return code.

FATS201 **TAPEx MAJOR ERROR ON TAPE**
ANALYZE TERMINATED FOR VOL=vvvvvv

Reason: The first form of the message will be issued whenever unrecoverable I/O errors (other than data checks) occur on the indicated tape drive; it will be followed by message FATS202. The second form of the message appears when a threshold of major errors is exceeded or when the error is too severe to continue (such as equipment check or intervention required).

Action: If the second form of the message is issued, further processing of the tape volume "vvvvvv" is halted, and FATS will eventually abend with a U0888 abend code to indicate that major errors occurred.

FATS202 **IOB SENSE ECB CSW CCW ADDRESS DCB FAILING CCW**
data corresponding to the above labels
FULL SENSE=ff....ff

Reason: This message is issued after message FATS201 when major I/O errors occur. At least the first two lines will be printed, the first containing the titles shown and the second the corresponding diagnostic data in hexadecimal. If the complete sense data produced by the device is available, it will be printed in the third line.

Action: Major errors may indicate severe problems with the tape being processed or problems with the tape unit being used. Appropriate manufacturer's manuals should be consulted to determine the cause of the error. Contact Innovation if you need assistance.

FATS203 **A line of the detail report indicating a tape mark read.**

FATS204 **A line of the detail report indicating a data check.**

FATS205 **A line of the detail report indicating PRINT threshold met.**

FATS206 **A line of the detail report showing the compaction achieved for the current line.**

FATS207 **A line of the detail report indicating operator cancelled, maximum errors exceeded, end-of-tape (tape indicate), or other errors that prevented FATS from completing successfully.**

FATS208 **A line of the detail report indicating cleaning action.**

FATS209 **A line of the detail report indicating that an IDRC compacted file was sensed.**

FATS210 **FATS OPEN/CLOSE SUBTASK ABEND ON DD=TAPEx COMP CODE = Ssss Uuuuu**

Reason: The FATS sub-task that handles OPEN, CLOSE, and FATAR has abended with a system (sss) or user (uuuu) abend.

Action: Use of the tape drive on TAPEx will be terminated. If no other TAPEx DD statements are active, FATS will terminate.

FATS211 **OPERATOR CANCELLED ANALYSIS ON DD=TAPEn**

Reason: Processing of this tape was cancelled by the operator by a reply of "KEOJ" or "Kuuu" to the FATSW02 console message, or "TERM" or "SKIP" to the FATSW08 console message.

FATS212 **TAPEx BLP OR NL NOT ALLOWED FOR ENCRYPTED TAPES. VOL=volser WILL NOT BE ERASED.**

Reason: An ERASE operation was requested and NL or BLP was specified on the TAPEx DD, but FATS doesn't support the use of NL or BLP for encrypted drives. Refer to the paragraph titled Erasing Encrypted Tapes in Section 11.5 for the guidelines to follow when erasing encrypted tapes.

FATS213 **TAPEx FILE NUMBER GREATER THAN 1 REQUIRED FOR ENCRYPTED TAPES. VOL=volser WILL NOT BE ERASED.**

Reason: An ERASE RESIDUAL operation was requested and SL or AL was specified on the TAPEx DD. FATS requires a file sequence number greater than 1 when erasing residual data from encrypted tapes. Refer to the paragraph titled Erasing Encrypted Tapes in Section 11.3 for the guidelines to follow when erasing residual data from encrypted tapes.

FATS214 **TAPEx BLP FORCED FOR RESIDUAL ERASE.**

Reason: ERASE RESIDUAL was specified, but the LABEL= parameter on the corresponding TAPEx DD contained a value other than BLP. BLP was forced to prevent the tape from being overwritten by the OPEN routines.

FATS215 ****RMM UPDATE RETRIED nnn TIMES FOR DSN=dsname ON VOL=volser**

Reason: RMMFAIL= RETRY was specified; an attempt to update the RMM information for a data set failed and was retried “nnn” times without success.

Action: The FATSCOPY job will abend.

FATS216 **DISP=(,KEEP) REQUIRED ON TAPEOUT DD**

Reason: When using TLMS, the TAPEOUT DD statement must contain DISP=(NEW,KEEP) (or DISP=(OLD,KEEP) when appending to an existing volume).

Action: The FATSCOPY job will terminate with a U100 abend. Add DISP=(NEW,KEEP) if writing to a new scratch tape; add DISP=(OLD,KEEP) if adding files to a non-scratch volume.

FATS217 **NO PHYSICAL VOLUMES FOUND FOR LOGICAL VOLUME volser**

Reason: The ALLPHYS keyword was used, but the specified logical volume was not found on any physical volume in your virtual tape system.

Action: The FATSCOPY job will terminate with a U300 abend. Be sure to specify a correct value for VIRTTYPE=, and a valid logical (not physical) volser as a parameter for the ALLPHYS= keyword.

FATS250 ****NOT ENOUGH STORAGE FOR TLMS UPDATE WORK AREA.**

Reason: FATSCOPY could not obtain the storage it needed to perform the update of the output file’s information in TLMS.

Action: The file(s) will be copied, but the tape management information will not be copied from the input file to the output file. A FATS148 message will also be issued. Rerun the job using REGION=0M.

FATS251 ****tapemgt INFORMATION UPDATED AFTER IMAGE COPY FOR VOLUME=volser**

Reason: FATSCOPY has done a successful image copy of the volume and updated tape management information for the volume. CA-1, RMM, or TLMS will be substituted for **tapemgt** in the message text.

Action: For RMM, the following fields have been updated in the volume record: assigned date, assigned time, vendor, compaction, location, volume capacity, storage group, media name, recording format, last read date (set from input volume before copying), last write date (set from input volume before copying), key label 1 and 2, and key encoding 1 and 2.

For CA 1, the following fields have been updated in the volume record: recording technique, last used unit address, temporary and permanent read and write errors since initialized, temporary and permanent read and write errors since last cleaning, date of last cleaning, use count at last cleaning, vendor name, encryption key index, percentage of volume in use, compression ratio, SMS management class, robotic device indicator, last used date (set from input volume before copying, and selected flags.

For TLMS, the following fields have been updated in the volume record: purchase date, tape type, manufacturer, length, out-of-service information, stacking status, box/cabinet ID, cleaning date and counts, read and write error counts, certification date and counts, location, move date, density, and auto tape library name.

FATS252 ****VOLUME=volser MOVED TO STORAGE GROUP storgrp.**

Reason: After updating the RMM information after a successful image copy of a volume in one storage group to a volume in another storage group, FATSCOPY issued an RMM command to move the volume to the new storage group. The TCDB has also been updated with a volume entry in the new storage group.

FATS253 ****tapemgt INFORMATION UPDATED AFTER IMAGE COPY FOR DATA SET=dsname VOL=volser**

Reason: After a successful image copy of a data set, FATSCOPY has updated tape management fields for the data set. RMM or TLMS will be substituted for **tapemgt** in the message text.

Action: For RMM, the following fields have been updated in the Data Set Record: last read date (set from input volume before copying), last write date (set from input volume before copying), and storage class.

For TLMS, the following fields (all set from the input volume before copying) have been updated in the Data Set Record: last used job, last used date, and last used unit.

FATS254 ****reason CA-1 UPDATE AFTER IMAGE COPY FOR VOLUME=volser FAILED.**

Reason: **reason** will be either CA-1 NOT ACTIVE or TMC OPEN FAILED. A FATS149 message showing the CA 1 reason codes will follow the FATS254 message.

Action: FATSCOPY will set a return code '8' for the failed update and continue copying the next volume. The volume record in CA 1 needs to be manually updated.

FATS255 ****CONTROL CARD ERROR – REASON x**

Reason: A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS255 message will be printed for each input statement (multiple errors will not be diagnosed). "x" is a character indicating the type of error detected:

2 – SELECT CATDSN specified with IMAGE keyword.
3 – RENAME statement specified with IMAGE keyword.
4 – A TMSIN value was not set by the FATZAPOP installation program, or an invalid combination of TMSIN and TMSOUT values has been specified.
5 – NUMVOLS= specified, but the volser specified on the SELECT ALLDSN statement contained no numeric characters.
6 – NUMVOLS= specified with SELECT CATDSN.
7 – Conflicting FILESEQ= or DSSIZE= keywords specified on a SELECT statement.
8 – Invalid value specified with the LOGERROR= keyword. Valid values are CONTINUE and ABEND.

Action: FATSCOPY will abend with a U0502 abend code after all control statements have been scanned.

FATS256 ****EOT ON TAPEOUT DURING IMAGE COPY, DSN=dsname VOL=volser. COPY TERMINATED.**

Reason: When doing an image copy of a volume, FATSCOPY reached the physical end of the output volume before copying all the data from the input volume.

Action: FATSCOPY will stop copying the input volume. If the input volume is part of a multi-volume set, FATSCOPY will skip any other volumes in the set, and continue copying any other volumes that remain. The job will abend with a U0888 after all the available volumes have been copied..

FATS257 ****VOLUME=volser NOT MOVED. ALREADY IN CORRECT STORAGE GROUP.**

Reason: After updating the RMM volume record following an image copy, FATSCOPY determined the input volume and the output volume were in the same storage group and no updates were necessary.

FATS258 ****WARNING** FOLLOWING OPTIONS CHANGED FOR IMAGE COPY:**

Reason: IMAGE was specified, but some of the options in the FATSCOPY Option Table or values specified on keywords are not allowed for image copies.

Action: FATSCOPY will print a line following the FATS258 message for each of the options/keywords that were changed because IMAGE was specified.

FATS259 ****UNABLE TO UPDATE TLMS INFO FOR VOLUME=volser AFTER IMAGE COPY. CODE=cccccccc, RC=rrrrrrrr**

Reason: FATSCOPY performed an image copy for the specified volume, but was unable to update device-specific TLMS information for the volume to show that it now resides on a different device. cccccccc and rrrrrrr can be used by Innovation to diagnose the cause of the problem.

Action: FATSCOPY will set a return code '8' for the failed update and continue copying the next volume. The volume record in TLMS needs to be manually updated. Contact Innovation for assistance in determining the cause of the update failure.

FATS260 ****RECATALOG OF DATA SET FAILED. COMP=comp,DSN=dsname**

Reason: During an image copy, FATSCOPY catalogs each copied file to the new output device. When an image copy of a volume fails, FATSCOPY recatalogs all copied files on that volume (or in that multivolume set) back to the original input device since the output tape may not be usable.

Action: The specified data set was not successfully recataloged. "comp" is the return code from CAMLST CATALOG. You will need to use IDCAMS to catalog the data set to the original input device.

FATS261 ****INPUT DATA SET RECATALOGED AFTER FAILED IMAGE COPY. DSN=dsname**

Reason: During an image copy, FATSCOPY catalogs each copied file to the new output device. When an image copy of a volume fails, FATSCOPY recatalogs all copied files on that volume (or in that multivolume set) back to the original input device since the output tape may not be usable.

Action: The specified data set was successfully recataloged.

FATS262 ****VOLUME volser BYPASSED FOR IMAGE COPY. NOT ALL DATA SETS SELECTED.**

Reason: One or more data sets on the specified volume did not meet the selection criteria. An image copy requires that all data sets on a volume be copied.

Action: The specified volume will not be copied. This message will be repeated for each volume in a multivolume set.

FATS263 ****FATSCOPY IMAGE COPY NOT SUPPORTED FOR OUTPUT DEVICE TYPE=xxxxxxxx. JOB TERMINATED.**

Reason: An image copy was attempted using an output device which is not supported for image copies. xxxxxxxx is either a hex device type printed as characters, or a text string describing the device.

Action: FATSCOPY will terminate with a U0888 abend.

FATS264 ****EXPDT=98000 NOT ALLOWED ON TAPEOUT DD FOR FATSCOPY IMAGE COPIES. EXPDT VALUE IGNORED.**

Reason: EXPDT=98000 was specified on the TAPEOUT DD when the IMAGE keyword was specified.

Action: FATSCOPY will ignore the expiration date specified on the TAPEOUT DD and assign the output expiration using the expiration date for the input volume from tape management, or the current date + 2 if the expiration date has passed.

FATS266 ****SYSTEM LOGGER FUNCTION FAILED. RETCODE=*retcode*, REAS=*reascode*,FUNCTION=*xxxxxxx* ….**

Reason: An attempt to use the system logger to write an audit record encountered an error.

Action: Be sure that a valid system logger file name has been specified in the AUDITLOG= parameter on the COPY statement, or in the AUDITLOG= value set in the FATSCOPY Global Options Table by the FATZAPOP utility program. The job will continue if LOGERROR=CONTINUE was specified; otherwise it will terminate. If a system logger file is not available for FATSCOPY, you can use an AUDIT DD statement to record audit information.

FATS267 ****LOGERROR=CONTINUE SPECIFIED. JOB WILL CONTINUE WITHOUT PRODUCING AUDIT RECORDS.**

Reason: A system logger error occurred while writing an audit record, and LOGERROR=CONTINUE was specified.

Action: The job continues, but audit records for data sets copied during the job will be missing or incomplete.

FATS268 ****LOGERROR=CONTINUE SPECIFIED. UNABLE TO DISCONNECT FROM LOGSTREAM AT END OF JOB.**

Reason: An error occurred when attempting to disconnect from the system logger at the end of the copy job.

Action: Contact Innovation for assistance.

FATS269 ****NO AUDIT RECORDS FOR DSN=dsname VOL=volser DUE TO SYSTEM LOGGER ERROR.**

Reason: The data set **dsname** was copied, but no audit record was produced for this data set due to an error writing to the system logger.

Action: The job will continue copying data sets, but may not produce any audit records for the data sets it copies.

FATS270 ****TCDB UPDATE FAILED FOR VOLUME=volser. RC=rtncode**

Reason: FATSCOPY attempted to update the TCDB after doing an Image Copy, but the update failed. The control statements used to update the TCDB and the error messages issued by Catalog Management will be printed after the FATS270 message.

Action: The job will abend with a U0888. Contact Innovation for assistance.

FATS271 ****OPEN OF SYSPRINT DATA SET FAILED. UNABLE TO PRINT CONTROL STATEMENTS AND ERROR MESSAGES.**

Reason: FATSCOPY attempted to open the data set that contained the control statements used to update the TCDB and the error messages issued by Catalog Management, but the open of the data set failed.

Action: Control statements and error messages will not be printed. The job will abend with a U0888.

FATS300 **A line of the summary report giving DD name, tape unit address, volume serial, operation performed, termination status, tape length, and data check summary by retry count.**

FATS301 **END OF REPORT**

Reason: Printed at the end of all detail and summary reports.

FATS345 **FILTER ERROR REASON=reason -- ENTRY=entry**

Reason: The SELECT CATDSN=filter was used to select entries from the system catalogs, and an error occurred. "entry" indicates the catalog name or the catalog entry on which the error occurred. "reason" indicates the error:

1 -- NO CATALOGED ENTRIES FOUND -- the filter did not select any entries from the catalogs.

2 -- CATLG ERROR COMP=*cccc* CODE=*reason* -- The IBM catalog SVC returned return code "xxxx" with reason code "reason". These codes can be found under message IDC3009I in IBM message manuals.

3 -- CATLG NAME FPL ADDR MISSING -- internal error.

4 -- CATLG VOLSER FPL ADDR MISSING -- internal error.

5 -- CATLG VOLSER 0 OR MORE THAN 20 -- a catalog entry was selected that had either 0 volsers or more than 20 volsers in a CVOL (SYSCTLG) catalog.

7 -- CATLG ENTYPE FPL ADDR MISSING -- internal error.

A -- NAME LAST CHARACTER HIGH VALUES -- internal error.

D -- CANNOT END IN A PERIOD -- The filter cannot end in a period. See Section 32.5 for filter rules.

E -- INVALID GENERATION NUMBER -- The filter ends in an invalid GDG relative generation number.

F -- SEQUENCE ERROR IN CATALOG -- A VSAM sequence error was encountered reading an ICF catalog, indicating a structural error in the catalog.

G -- ALL ENTRIES EXCLUDED OR DUPS -- all catalog entries selected by this filter were discarded because:

1) they were excluded by a preceding EXCLUDE statement
2) they were previously selected by a preceding SELECT CATDSN= statement
3) none were found on the specified VOL or DEVTYPE

J -- INSUFFICIENT STORAGE – the REGION size was not large enough. Increase the REGION (using REGION=0M is recommended) and resubmit.

Action: Serious errors will cause a U0502 ABEND.

FATS346 **FILTER SELECTED nnnnn ENTRIES IN sssss.ss SECONDS**

Reason: This describes the number of entries selected by the SELECT CATDSN=filter, and the elapsed time in seconds required to complete the search.

FATS400 ***program*/*function*** **-- VER *v.r*/*mmt* -- INNOVATION DATA PROCESSING DATE=*yyyy.ddd* TIME=*hh.mm.ss* PAGE *nnnn***

Reason: General page header message for FATZAPOP programs. The time is not reported by all programs.

FATS401 **PARAM DATA -- * *parm-field-data* ***

Reason: Displays the program control information specified in the 'PARM=' field of the EXEC statement.

FATS402 **INVALID CONTINUATION**

Reason: The last FATZAPOP control statement ended in a comma, but there was no next statement or it was not a proper continuation statement.

Action: Correct and resubmit job.

FATS405 **MAXIMUM CONTINUATION COUNT OF *nnnn* EXCEEDED -- COMMAND FLUSHED**

Reason: A FATZAPOP control statement used too many continuation statements.

Action: Combine multiple operands onto one line to reduce the number of continuations.

FATS407 ***type* ERROR -- *action***

Reason: An error was encountered during the processing of FATZAPOP control statements. This is always proceeded by one or more messages that define and delimit the error(s). "***type***" will be either **PARAMETER FIELD** or **CONTROL STATEMENT**.

The "*action*" is one of the following:

JOB TERMINATED – Processing will stop after the first error has been encountered.

SKIPPING FOR COMMAND – Processing will continue for all command statements found within the SYSIN data set.

RE-ENTER COMMAND OR END – Message for a user that has SYSIN data set assigned to a TSO terminal. Re-enter command in error or '**END**' to complete the processing.

Action: Correct and resubmit job.

FATS408 **NO CONTROL STATEMENTS WERE FOUND -- JOB TERMINATED**

Reason: The SYSIN data set is empty, or contained only comment statements ('*' in column 1). The first control statement can also be supplied by the EXEC statement PARM= option. There are other forms of this message that indicate that statements required for a certain function were not found.

Action: Correct and resubmit job.

FATS418 **COMMAND PROCESSING DETECTED ERROR--*action***

Reason: An error was encountered during the processing of a FATZAPOP statement. This is always proceeded by one or more messages that define the error. "action" will be one of:

SKIPPING FOR COMMAND – processing will continue for all command statements found within the SYSIN data set.

REVERTING TO SYSIN – the error occurred when reading from an alternate command input source. Processing will continue for commands in the SYSIN data set.

RE-ENTER COMMAND OR END – message for user that has SYSIN data set assigned to a TSO terminal. Re-enter command in error or '**END**' to complete the processing.

Action: Correct and resubmit job.

FATS465 **UNABLE TO OPEN DDNAME=*ddname--reason***

Reason: An error occurred trying to open "***ddname***", for the "***reason***" displayed.

Action: Check the joblog for IBM OPEN messages. In some cases, the DDname is not required so the message is simply informational. If the DDname is required for the operation being attempted, correct and resubmit the job.

FATS471 **DDNAME=*ddname* I/O ERROR--SYNAD=*synadtext***

Reason: A permanent I/O error was detected on the data set referenced by "***ddname***". The IBM macro SYNADAF was invoked to format the cause of the error; the text returned is "***synadtext***".

Action: Examine the SYNADAF text to determine the cause of the error. Check the joblog for IBM I/O error messages. For additional assistance, please call Innovation Data Processing.

FATS484 **INTERNAL LOGIC ERROR--JOB TERMINATED**

Reason: The program has encountered an illogical condition.
Action: Call Innovation Data Processing for technical assistance after obtaining a storage dump.

FATS491 ***function*** **FUNCTION STARTED--*hh.mm.ss***

Reason: Identifies type of FATZAPOP function and time the function started.

FATS492 ***function*** **FUNCTION ENDED--*hh.mm.ss* CONDITION CODE-*nnnn***

Reason: Identifies type of FATZAPOP function and time the function ended. The return code is printed if nonzero.

FATS493 ***function*--BYPASSED--PRIOR FUNCTION TERMINATED WITH KEYWORD OR COMMAND DETECTED ERROR**

Reason: A previous command on which this function may be dependent encountered serious errors.
Action: Correct the condition and re-execute.

FATS496 **MODULE *modname* NOT USABLE WITH RELEASE/SYNC LEVEL *nnn* PROGRAMS - EXECUTION TERMINATED**

Reason: The main FATZAPOP program determined that a subprogram was not at a compatible level, probably because different versions of the FATSCOPY program library are being used.
Action: Contact Innovation Data Processing for assistance.

FATS497 ***program*** **ABNORMALLY TERMINATED DUE TO KEYWORD/COMMAND DETECTED ERRORS**

Reason: "***program***" did not complete normally.

FATS498 ***program*** **PROCESSING COMPLETED WITH ERRORS**

Reason: "***program***" completed the required processing but non-terminating errors were encountered.
Action: Check the output, correct and rerun or restart as appropriate.

FATS499 ***program*** **PROCESSING COMPLETED**

Reason: "***program***" completed the required processing successfully.

FATS530 ***member*** **NOT FOUND--*ddname=dsname***

Reason: FATZAPOP issued a BLDL for the module "***member***" in the library "***dsname***" specified on the DD statement "***ddname***", but the module was not found. The module is required.
Action: Verify that the data set name specified on the DD statement was the correct one. If incorrect, change and resubmit the job. If the data set name is correct, check to see if all the installation steps required were done (see section 90). Contact Innovation Data Processing for technical assistance..

FATS531 **MODULE *member* CONTAINS NO TEXT RECORDS – *ddname=dsname***

Reason: Program attempted to read module "***member***" in the library "***dsname***" specified on the DD statement "***ddname***", but no text records were found. The module is required.
Action: See message FATS530.

FATS532 **MODULE *member* I/O ERROR READING--*ddname=dsname***

Reason: Program attempted to read module "***member***" in the library "***dsname***" specified on the DD statement "***ddname"***, but an I/O error occurred. The module is required.

Action: Other messages may be displayed to further define the I/O error. Check the joblog for IBM error messages. Contact Innovation Data Procesing for technical assistance.

FATS533 **MODULE *mmmmmmmm* I/O ERROR WRITING--*ddname=dsname***

Reason: Program attempted to update module "***mmmmmmmm"*** in the library "***dsname***" specified on the DD statement "***ddname***", but an I/O error occurred. The update was not completed, but the module might be partially updated. This is a serious error that may result in the load module library becoming unusable.

Action: Other messages may be displayed to further define the I/O error. Check the joblog for IBM error messages. Contact Innovation Data Processing for technical assistance.

FATS534 ***ddname* OPEN FAILURE--AVAILABLE COMMANDS: HELP, END**

Reason: The required library on DD statement "***ddname***" could not be opened. Commands that access modules in that library cannot be executed.

Action: Check the joblog for IBM OPEN error messages. Correct the error and rerun.

FATS535 ***mmmmmmmm* READ UNSUCCESSFUL--*function* BYPASSED**

Reason: Program attempted to read a member from a library or a data set but the read was not successful. "***mmmmmmmm***" can be:

- module MODULE – if the error occurred trying to read a load module.
- MEMBER – if the error occurred trying to read another kind of member.
- DATA SET – if the error occurred trying to read a sequential data set or member.

Action: See message FATS530.

FATS538 **MODULE *member* WRONG VER/LEVEL--*ddname=dsname***

Reason: Program read module "member" in the library "***dsname***" specified on the DD statement "***ddname***", but the module loaded is not at the same level as the calling program.

Action: Verify that the data set name specified on the DD statement was the correct one. If incorrect, change and resubmit the job. If the data set name is correct, check to see if all the installation steps required were done (see section 90). Contact Innovation Data Processing for technical assistance..

FATS539 **MODULE *member function--ddname=dsname***

Reason: The module ***"member***" in the library "***dsname***" specified on the DD statement "***ddname***" has been successfully processed. "function" indicates what operation was done.

FATS540 ***operand* CONTAINS INVALID CHARACTERS--ZAP REJECTED**

Reason: The value specified for "***operand***" contained one or more characters that were not A-Z, 0-9, $#@.

Action: Correct the value and resubmit the job.

FATS542 ***operand* CONTAINS INVALID INDEX STRUCTURE--ZAP REJECTED**

Reason: The value specified for "***operand***" contains two (2) or more consecutive periods (..) in violation of MVS data set naming conventions.

Action: Correct the error and resubmit the job.

FATS543 ***operand*** **DOES NOT CONTAIN CHARACTER STRING** ***rrrrrrr*****--ZAP REJECTED**

Reason: The value specified for "***operand***" does not contain the character string "***rrrrrrr***" as an index level.

Action: Provide a proper data set name and resubmit the job.

FATS544 ***operand*** **REJECTED--EXCEEDS MODIFIABLE PORTION OF FATOPT**

Reason: The length of the value specified for "***operand***" taken in conjunction with the offset operand exceeds the modifiable portion of FATOPT.

Action: Contact Innovation Data Processing for assistance.

FATS545 **VERIFY FAILED--CHAR/HEX PRINT FORCED**

Reason: The VERIFY of existing contents failed. A character/hex print of the module FATOPT is produced. Always proceeded by message FATS544.

FATS516 ***function*** **REGISTERS -- R0=xxxxxxxx R1=xxxxxxxx R15=xxxxxxxx**

Reason: "***function***" may be:

- CAMLST – A CAMLST request for a CATALOG operation or a SCRATCH failed.
- CATLG or CATALOG – A request to catalog a data set failed.
- LOCATE – A LOCATE request to read a catalog record failed.
- OBTAIN – An OBTAIN request to read a DSCB from the VTOC failed.
- RENAME – A request to RENAME a data set failed.
- SCRATCH – A SCRATCH request to delete a data set failed.
- UNCATLG – A request to uncatalog a data set failed. The contents of registers 0, 1, and 15 are displayed in hex; R15 will contain the return code but the other registers may have additional diagnostic data. The failing operation may be further described in a preceding message. The return codes are documented in the IBM manual DFSMSdfp Advanced Services, or under message IDC3009I in the System Messages manual.

Action: Check the return codes and take corrective action or call Innovation Data Processing for assistance.

FATS541 **ALLINDX CREATES** ***name*** **NAME GT nn CHARACTERS--ZAP REJECTED**

Reason: The value specified for the operand ALLINDX, when used to replace the first index of the option "***name***", creates a character string longer than the limit "***nn***".

Action: Shorten the value of ALLINDX by specifying fewer characters and resubmit the job. Some options may have already been changed before the message is issued.

FATS550 **ERRORS ENCOUNTERED DURING EXECUTION--REWRITE CANCELLED**

Reason: One or more modules from the SYSLIB data set have been updated and were scheduled to be rewritten to the SYSLIB data set. However, previous commands failed to complete successfully so the updates were discarded and the modules were not modified.

Action: Correct the error conditions documented by the error messages and resubmit the job.

FATS569 **WARNING --** ***option*** **NOT CATALOGED--DSN=***dsname*

Reason: The FATSCOPY Global Options Table was updated with a new data set name. The new data set name was not in the system catalogs and may not exist.

Action: This is only a warning message. The data set name listed has become the new name of the indicated default.

FATS800 **LOGGER *errortype* ERROR. RC=*retcode*, REAS=*reascode***

Reason: FATAUDIT encountered an error communicating with the system logger while attempting to access the audit log stream. ***errortype*** will be one of the following: **CONNECT**, **START**, **READCURSOR**, **DEBLOCKING**, **DISCONNECT**, **RECONNECT**, or **PURGE**. This will be accompanied by a U04nn abend.

Action: Contact Innovation Technical Support for assistance.

FATS804 **INVALID LOG STREAM NAME = *streamname***

Reason: FATAUDIT could not open a system logger log stream with the specified stream name. This will be accompanied by a U0404 abend.

Action: Verify that the ***streamname*** value matches the name used to start the FATSCOPY audit log stream.

FATS805 **EMPTY INPUT LOG STREAM = *streamname***

Reason: FATAUDIT did not find any data in the specified log stream. A FATAUDIT PURGE may have been done without any subsequent copy jobs writing to the log stream. This will be accompanied by a U0805 abend.

FATS990 **INSTALLED INNOVATION TRIAL WILL EXPIRE ON yy.ddd**

Reason: Product extension has completed successfully.
Required JCL:

```
//EXTEND  EXEC  PGM=FATEXTND,PARM=xxxx
//STEPLIB DD DISP=SHR,DSN=fats.loadlib
//SYSLIB  DD DISP=SHR,DSN=fats.loadlib
//SYSDIAG DD SYSOUT=*
```

The PARM= value will be supplied by Innovation.

40.5 FATS ACTION MESSAGES

ACTION MESSAGES

The "ACTION" column of the FATS detail report will contain one of the following messages:

3480 FORMAT FILE SENSED

The current file is a normal 3480 file.

3480XF FORMAT FILE SENSED

The current file is an IDRC compacted 3480 file.

3490E FORMAT FILE SENSED

The current file is a 3490E format file that may or may not be IDRC compacted.

3590 FORMAT FILE SENSED

The current file is a 3590 (Magstar) format file that may or may not be IDRC compacted.

DATA COMPACTED BY xx%

The current file was IDRC compacted by the indicated amount.

TAPE INDICATE

The logical end of tape (end-of-tape reflector or cartridge logical end-of-tape) has been detected by a FATS WRITE operation.

PERM DATA CHECK

An error has been retried until the retry level defining a permanent error (RETRY= parameter) has been reached.

PERM DATA CHECK CNTL

An error was encountered while repositioning the tape for data check retry. No further retries are done. This may indicate serious problems with the tape or drive.

TEMP DATA CHECK

An error has been retried and was successful before the permanent error retry level was reached.

TAPE MARK

A tape mark (end of file) was read during a FATS read operation.

LABEL SAVED

The volume serial was read from the tape and labels were successfully rewritten (the SAVLAB function).

LABEL NOT SAVED

The SAVLAB function was attempted, but labels did not exist or could not be read.

LABEL WRITTEN

Labels with the specified volume serial number have been written to the tape.

LABEL NOT WRITTEN

Labels could not be written due to I/O errors.

TEMP DATA CHK LABEL

Temporary I/O errors were encountered when writing labels but the labels were written successfully.

PERM DATA CHK LABEL

Permanent I/O errors were encountered when writing or saving labels. The labels were not written.

MAX ERROR EXCEEDED

The length in inches of a contiguous data check has exceeded the MAXCERR= parameter specified. Processing of this tape will be terminated.

PRINT THRESHOLD MET

The number of errors printed for this tape exceeds the THRESHOLD= parameter specified. No more errors will be printed in the detail report, but processing will continue.

OPERATOR CANCELLED

Processing of this tape was cancelled by the operator by a reply of "KEOJ" or "Kuuu" to the FATSW02 console message or "TERM" or "SKIP" to the FATSW08 console message.

OPEN ERROR OR NO DD

Errors occurred opening this tape. Processing of all tapes on this tape unit will be halted.

CLEAN ACTION REQUEST

The number of errors on this tape exceeds the ERRCLEAN= parameter. The operator is requested to clean the tape drive and remount the same tape.

This page intentionally left blank.

90.0 FATS/FATAR/FATSCOPY INSTALLATION

The installation of FATS/FATAR and FATSCOPY can be done in 2 different ways. **Do not read all of this chapter;** follow the path for the installation method that you wish to use.

Electronic Installation – The install package is downloaded from the Innovation FTP site or received via an email. If you are installing the products using this method, **follow the instructions in Section 90.1.1, Steps 1.1 through 1.6, starting on the next page**.

CD Installation – The install package is downloaded from a CD. If you are installing the products using this method, **follow the instructions in Section 90.1.2. Steps 2.1 through 2.4, starting on page 90-8.**

90.1 LOADING THE FATS/FATAR/FATSCOPY LIBRARIES

90.1.1 ELECTRONIC INSTALLATION

STEP 1.1 For an electronic installation, instructions will be sent in a set of 2 emails for installing the libraries. **The email installation instructions that you receive supersede these printed instructions** and should be used if any differences are noted.

The first email, the order acknowledgement, will look similar to:

Subject: Innovation Distribution Server Order Acknowledgment 012345

From: ENSMTP@CPUA.IDPNJ.COM

Date: Mon., 18 November 2013 09:09:18

To: SYSPROG@ABCCOMPANY.COM

You are receiving the first of a two message set acknowledging your Innovation software product order. Retain this e-mail until you receive a status (second) message containing instructions for performing the product installation. The status message will be sent when your order is ready to be downloaded from the Innovation FTP site.

NOTE:

1. You must perform the install of this distribution file
   before TUESDAY DECEMBER 3 2013.
   The install program IDPREC in your distribution file
   will expire on this day and the file will be removed from our FTP site.

This order is for FAT Product Group including the following options:
FAT FTR FTC

Customer Order Number:
012345012345012345
Company name:
ABC COMPANY
Customer contact:
J. SYSPROG
Product version:
4.9.28
IDPREC Password:
0909090909090909

This email contains the products and versions that will be included in your installation package, the expiration date (if the product has an expiration date), and a unique IDPREC password that is required for the installation process.

STEP 1.2 The second email, the order completion, will look similar to:

Subject: Innovation Distribution Server Order Completion 012345

From: ENSMTP@CPUA.IDPNJ.COM

Date: Mon, 18 November 2013 09:10:09

To: SYSPROG@ABCCOMPANY.COM

You are receiving the second of a two message set acknowledging your Innovation software product order is ready to be downloaded from the Innovation FTP site. Verify that the Customer Order Number is the same on both e-mail messages before performing the retrieval procedure below.

This order is for FAT Product Group including the following options:
FAT FTR FTC

Customer Order Number:
012345012345012345
Customer number:
012345
Company name:
ABC COMPANY
Customer contact:
J. SYSPROG
Product version:
4.9.28
IDPREC Password:
0909090909090909

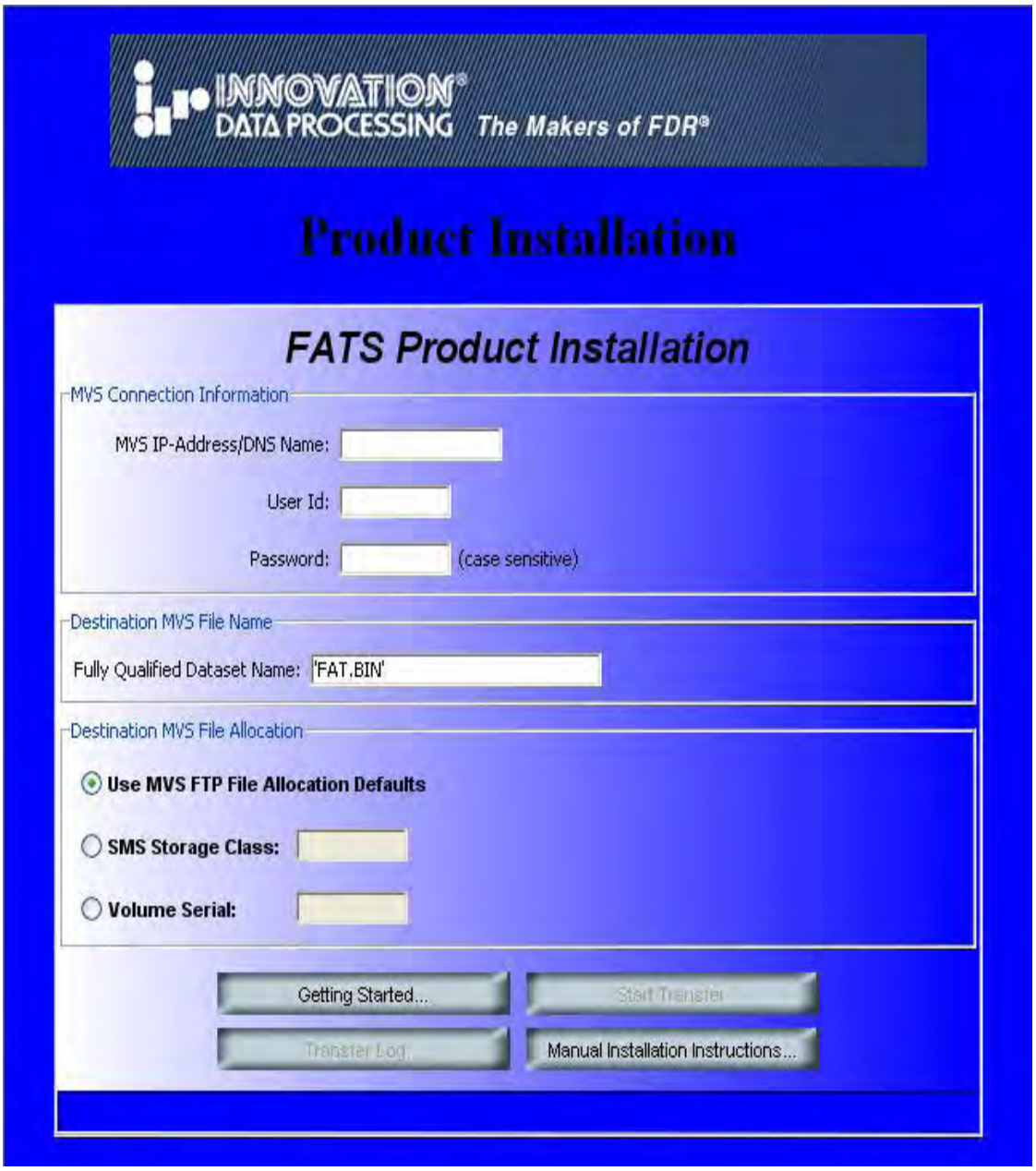
The instructions below outline the procedure you will use to retrieve the product file from the Innovation FTP site and then install our software.

The most convenient method is to **. . .**

This email contains the links to a unique FTP site containing the product installation package. The transfer of the installation package can be done with a Java applet, or by manual FTP download if you encounter problems running the Java applet at your site. Instructions are included in this email for both methods.

STEP 1.3 The Java applet installation procedure is the easiest and will download a product distribution file from the Innovation FTP site to your PC using the Windows FTP.EXE client, It will then connect to your z/OS FTP server to transfer this file to z/OS. The transfer from the PC to z/OS is done within the user's local network. You will need to supply your z/OS connection information and click on the button labeled "Start Transfer".

Upon successful file transfer to your z/OS host, the applet will display a confirmation message box labeled "Transfer to z/OS Host Successful". After clicking "OK", another message box containing the remaining z/OS installation instructions will be displayed.



STEP 1.4 Follow the “Remaining Product Installation Instructions” that are presented at the end of the file transfer. Expanding the product distribution file on your z/OS host is a two step process. Log on to your z/OS host and enter a TSO READY session (outside of ISPF). This TSO session must have access to an ISPF environment because the z/OS install process expects to employ ISPF panels to complete the installation. The steps described below assume TSO PROFILE NOPREFIX is NOT used.

1. `RECEIVE INDATASET(FAT.BIN)`

   where the INDATASET(…) value is the name that you created during the file transfer to z/OS.

   This creates a partitioned data set containing two load modules. After the RECEIVE is completed, the name of the PDS will be ‘userid.IDPREC.LOAD’. If you want a different name, then at the prompt

   ```
   Data set FATSYS.IDPREC.LOAD from FATCFG on JESCPUA
   Enter restore parameters or 'DELETE' or 'END' +
   ```

   Enter:

   ```
   DSN(desired.name)
   ```

   Or take the default by pressing ENTER.

2. `CALL 'userid.IDPREC.LOAD(IDPREC)'`

   or, if you gave the PDS a different name:

   ```
   CALL 'dsname(IDPREC)'
   ```

   A welcome message is displayed. Enter appropriate responses to name and sysout class prompts.

   You will also be prompted to enter the 16 digit IDPREC PASSWORD supplied in the Order Acknowledgement email.

   Follow the instructions on the ISPF panel to complete the installation of the distribution files.

STEP 1.5

Screen 1: Data Set Name Selection

This screen allows you to specify the data set names that will be used for the data sets you are loading from the installation package. These may be existing data sets to be updated, or they may be new data sets that will be allocated and cataloged; new data sets are recommended. Do not install FATS/FATAR and FATSCOPY into existing libraries containing other FATS products, since they may share common modules.

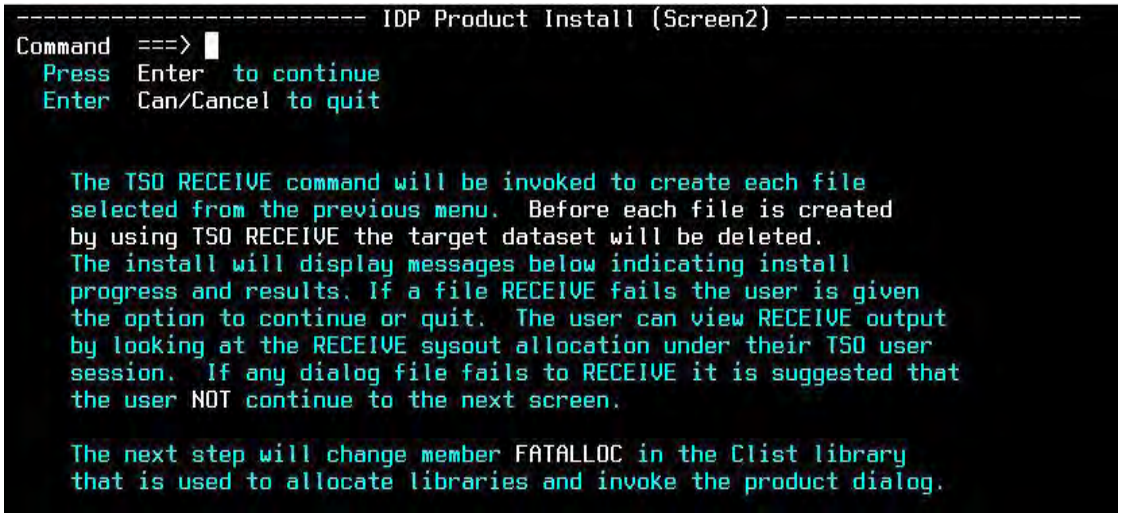
The load library must be an APF authorized library. If necessary, you can authorize it after the install using the SETPROG console command; see the IBM “System Commands” manual for details.

The names shown above are the default names provided by the IDPREC Install program, with the default name of IDP being used. You may change these names in one of 2 ways:

1. To change the High or 2nd Level qualifiers for all data sets, change the values in the corresponding field for the Output DSN section.
2. To change an individual Low Level qualifier, change desired Low Level Qualifier for the data set(s) that you want to change.

The names resulting data set names that will be used are displayed at the bottom area of the panel.

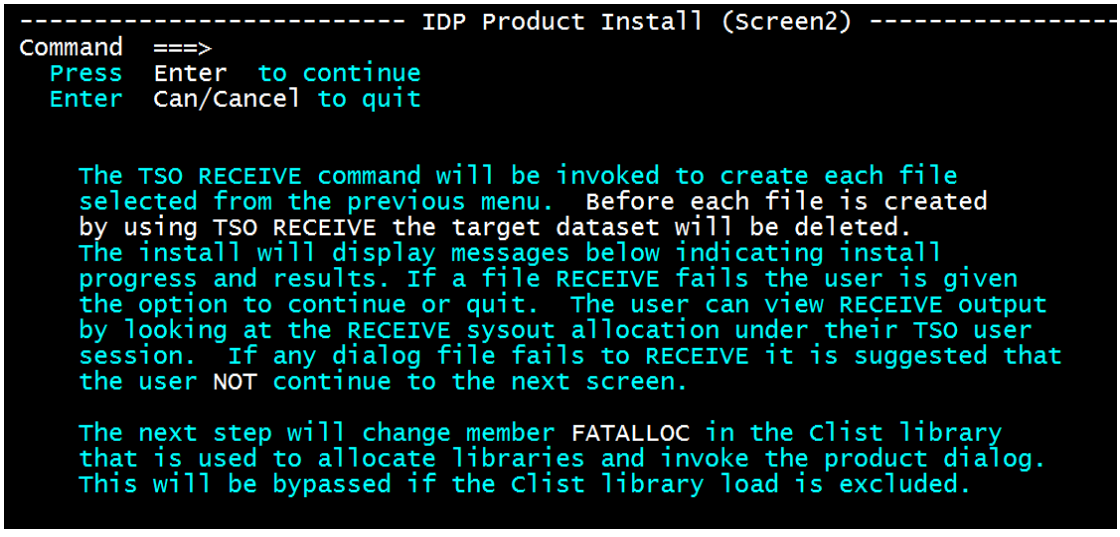
You can also specify the SMS attributes (if they are to be SMS-managed) or VOLUME/UNIT information on this panel to allocate these data sets when creating them as NEW.


IDP Product Install (Screen2)
Command ===>
Press Enter to continue
Enter Can/Cancel to quit
The TSO RECEIVE command will be invoked to create each file
selected from the previous menu. Before each file is created
by using TSO RECEIVE the target dataset will be deleted.
The install will display messages below indicating install
progress and results. If a file RECEIVE fails the user is given
the option to continue or quit. The user can view RECEIVE output
by looking at the RECEIVE sysout allocation under their TSO user
session. If any dialog file fails to RECEIVE it is suggested that
the user NOT continue to the next screen.
The next step will change member FATALLOC in the Clist library
that is used to allocate libraries and invoke the product dialog.

Screen 2: IDP Product Install

This screen documents the process that will be used to create the product libraries. When you press ENTER to continue, it will also show you the status of each of the data sets being created. Once all the product libraries have been created, a completion message will be displayed.

STEP 1.6


IDP Product Install (Screen2)
Command ===>
Press Enter to continue
Enter Can/Cancel to quit
The TSO RECEIVE command will be invoked to create each file
selected from the previous menu. Before each file is created
by using TSO RECEIVE the target dataset will be deleted.
The install will display messages below indicating install
progress and results. If a file RECEIVE fails the user is given
the option to continue or quit. The user can view RECEIVE output
by looking at the RECEIVE sysout allocation under their TSO user
session. If any dialog file fails to RECEIVE it is suggested that
the user NOT continue to the next screen.
The next step will change member FATALLOC in the Clist library
that is used to allocate libraries and invoke the product dialog.
This will be bypassed if the Clist library load is excluded.

Screen 3: Installation Data Sets Have Been Received

This screen documents that all the product libraries have been created and received.

Proceed to Section 90.3 on page 90-12 to continue the installation.

90.1.2 INSTALLATION FROM CD

STEP 2.1 The CD installation method contains a letter that documents the install process. (A copy of this letter is in electronic form in the Readme.rtf file in the z/OS directory found in the root directory of the CD). You should follow these instructions provided to install the product using the CD install method. **The letter's installation instructions supersede these printed instructions** and should be used if any differences are noted.

You must transfer the FAT.BIN file from the z/OS directory found in the root directory of the CD to your z/OS host. The required file attributes for the target z/OS data set will be listed in the instructions. The exact directions for the FTP transfer are also documented in the instructions.

STEP2.2 Once the FAT.BIN file is on the z/OS host system, you will then expand the product distribution file. This is a two-step process requiring you to logon to your z/OS host and enter a native TSO session (exit ISPF). This TSO session must have access to an ISPF environment because the z/OS install process expects to employ ISPF panels to complete the installation. The steps described below assume TSO PROFILE NOPREFIX is NOT used.

1. `RECEIVE INDATASET(FAT.BIN)`

   Where the INDATASET(…) value is the name that you created during the file transfer to z/OS.

   This creates a partitioned data set containing two load modules. After the RECEIVE is completed, the name of the PDS will be 'userid.IDPREC.LOAD'. If you want a different name, then at the prompt

   ```
   Data set FATSYS.IDPREC.LOAD from FATCFG on JESCPUA
   Enter restore parameters or 'DELETE' or 'END' +
   ```

   Enter:

   ```
   DSN(desired.name)
   ```

   Or take the default by pressing ENTER.

2. `CALL 'userid.IDPREC.LOAD(IDPREC)'`

   or, if you gave the PDS a different name:

   ```
   CALL 'dsname(IDPREC)'
   ```

   A welcome message is displayed. Enter appropriate responses to name and sysout class prompts.

   You will also be prompted to enter the 16 digit IDPREC PASSWORD supplied in the Order Acknowledgement email.

   Follow the instructions on the ISPF panel to complete the installation of the distribution files.

STEP 2.3

Screen 1: Data Set Name Selection

This screen allows you to specify the data set names that will be used for the data sets you are loading from the install package. These may be existing data sets to be updated, or they may be new data sets that will be allocated and cataloged; new data sets are recommended. Do not install FATS/FATAR and FATSCOPY into existing libraries containing other FATS products, since they may share common modules.

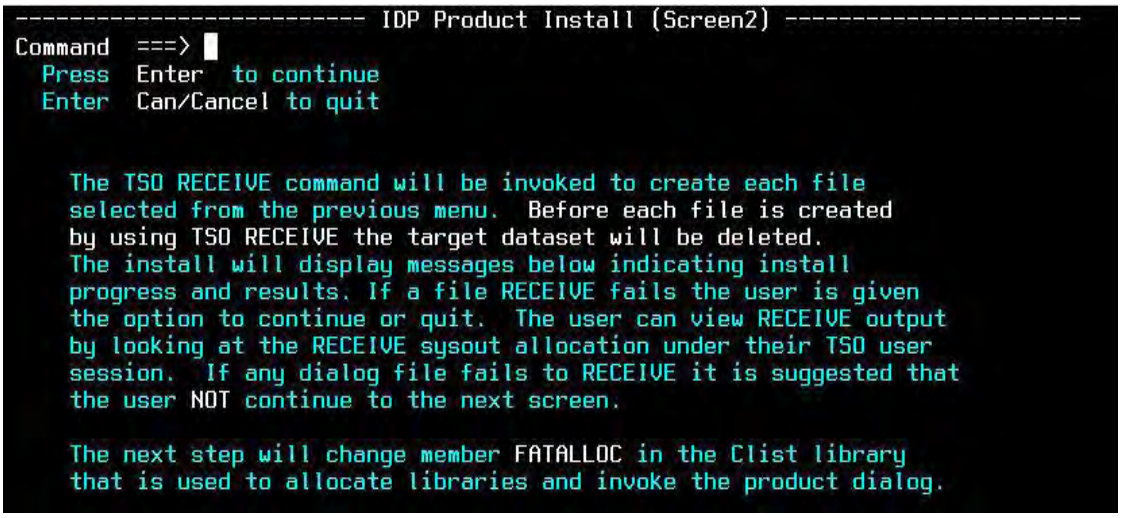
The load library must be an APF authorized library. If necessary, you can authorize it after the install using the SETPROG console command; see the IBM “System Commands” manual for details.

The names shown above are the default names provided by the IDPREC Install program, with the default name of IDP being used. You may change these names in one of 2 ways:

1. To change the High or 2nd Level qualifiers for all data sets, change the values in the corresponding field for the Output DSN section.
2. To change an individual Low Level qualifier, change desired Low Level Qualifier for the data set(s) that you want to change.

The names resulting data set names that will be used are displayed at the bottom area of the panel.

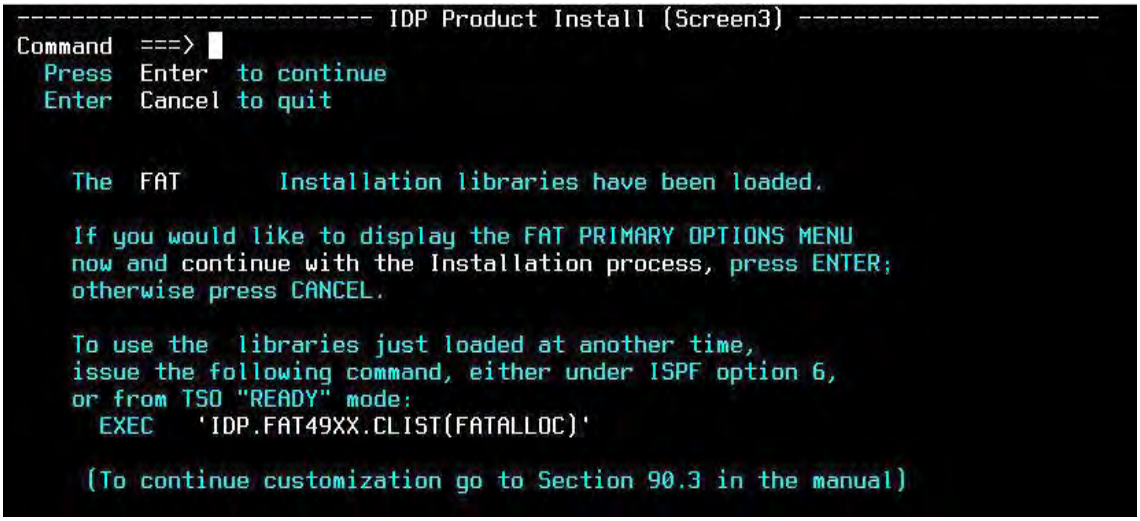
You can also specify the SMS attributes (if they are to be SMS-managed) or VOLUME/UNIT information on this panel to allocate these data sets when creating them as NEW.


IDP Product Install (Screen2)
Command ===>
Press Enter to continue
Enter Can/Cancel to quit
The TSO RECEIVE command will be invoked to create each file
selected from the previous menu. Before each file is created
by using TSO RECEIVE the target dataset will be deleted.
The install will display messages below indicating install
progress and results. If a file RECEIVE fails the user is given
the option to continue or quit. The user can view RECEIVE output
by looking at the RECEIVE sysout allocation under their TSO user
session. If any dialog file fails to RECEIVE it is suggested that
the user NOT continue to the next screen.
The next step will change member FATALLOC in the Clist library
that is used to allocate libraries and invoke the product dialog.

Screen 2: IDP Product Install

This screen documents the process that will be used to create the product libraries. When you press ENTER to continue, it will also show you the status of each of the data sets being created. Once all the product libraries have been created, a completion message will be displayed.

STEP 2.4


IDP Product Install (Screen3)
Command ===>
Press Enter to continue
Enter Cancel to quit
The FAT Installation libraries have been loaded.
If you would like to display the FAT PRIMARY OPTIONS MENU
now and continue with the Installation process, press ENTER;
otherwise press CANCEL.
To use the libraries just loaded at another time,
issue the following command, either under ISPF option 6,
or from TSO "READY" mode:
EXEC 'IDP.FAT49XX.CLIST(FATALLOC)'
(To continue customization go to Section 90.3 in the manual)

Screen 3: Installation Data Sets Have Been Received

This screen documents that all the product libraries have been created and received.

Proceed to Section 90.3 on page 90-12 to continue the installation.

90.2 ALTERNATE MANUAL INSTALLATION

If you prefer not to use the recommended interactive installation program, you can load the libraries to disk manually.

JCL FOR LOADING INSTALL JOBSTREAM

First, copy the installation job stream from the ICL (Installation Control Library) on tape to disk, using JCL similar to:

```
//INSTALL   JOB   ...
//COPY   EXEC  PGM=IEBCOPY
//SYSPRINT  DD    SYSOUT=*
//TAPEIN    DD    DSN=ICL,DISP=OLD,LABEL=4,UNIT=tape,
//            VOL=SER=FAT49T << change to FAT49P if production tape
//LIBRARY   DD    DSN=your.library.here,
//            DISP=SHR
//SYSIN     DD    *
 COPY INDD=((TAPEIN,R)),OUTDD=LIBRARY
 SELECT MEMBER=INSTALL
//*
```

You will need to specify an appropriate unit name for the tape; if your tape is a production tape, be sure and change the tape volume serial to FAT49P. The LIBRARY DD statement should point to a control statement library (RECFM=FB and LRECL=80); this can be an existing library or you can allocate a new one. The job stream above will load **only** the single member **INSTALL**.

If you want to **only** install the FATS/FATAR/FATSCOPY Load Library, use the member **INSTMIN**.

INSTALLATION JCL

The JCL member loaded by the preceding job provides you with procedures for the installation of the FATS/FATAR/FATSCOPY:

- Product load library
- Installation Control library
- JCL (example) library
- ISPF dialog libraries

The JCL stream consists of three in-stream procedures and the necessary steps to execute those procedures to install the FATS/FATAR/FATSCOPY libraries. One procedure allocates the target libraries and contain the recommended size of each library; the other two execute IEBCOPY or IEBUPDTE to load the library from tape to disk. The job stream contains comments that will guide you through modifying the job stream to substitute data set names and target disk volumes of your choice. Once modifications are complete, submit the job stream to allocate and load the libraries.

Note: The FATS/FATAR/FATSCOPY load library must be an authorized program library as shown in Section 90.1.

90.3 ACTIVATING THE TAPE MANAGEMENT INTERFACE

> **Please note: this procedure replaces** the tape management installation procedure used by FATAR and FATSCOPY in version 4.9.25 and previous versions. You no longer run an assembly job using member ASMCA1, ASMTLMS, or ASMRMM. Those members are no longer provided.

- **For FATSCOPY, installation of the tape management interface is REQUIRED**.
- The tape management interface is optional for FATAR, and is used by the FATAR ISPF dialogs to query tapes to be processed. Details on this interface are in Section 21.5.
- If you are licensed for only FATS, you will not use a tape management interface and can skip this section.

IBM's DFSMSrmm (RMM), and CA Technologies' CA 1 (TMS) and TLMS, are supported by FATAR and FATSCOPY.

ACTIVATING THE INTERFACE

There are two ways to activate the tape management interface: using the ISPF Dialog, or with a batch job.

If you are going to install the ISPF dialog, you may **proceed to section 90.5 to install it.**

Otherwise, to activate the tape management interface with a batch job, run the FATZAPOP program that is provided with FATAR and FATSCOPY. FATZAPOP (fully described in Section 90.4.1) is a program that can optionally be used to set many FATSCOPY defaults. However, to define your tape management system to FATAR or FATSCOPY, it must be run it at least once with the ZAP TMSIN= statement to define your tape management system to FATAR or FATSCOPY. **For FATSCOPY, this is a required step**.

Run the following job to activate the tape management interface. Substitute the name of your tape management system for "`tapemgt`". Use the ZAP VMFDSN= statement only if you are using TLMS tape management, using the name of your TLMS Volume Master File. Additional FATZAPOP command statements, described in Section 90.4.1, may be added to this job but are not required. (This job can be found in the EX900301 member of the JCL library.)

```
//ZAPOP     EXEC  PGM=FATZAPOP,REGION=0M
//STEPLIB   DD    DISP=SHR,DSN=fatsfatr.load.library
//SYSLIB    DD    DISP=SHR,DSN=fatsfatr.load.library
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD    *
 ZAP TMSIN=tapemgt              <- tapemgt = RMM, CA1, or TLMS
 ZAP VMFDSN=tlms.volume.master.filename   <- use only for TLMS
/*
```

TESTING THE INTERFACE

You can quickly test the tape management interface using a test program, FATTMTST. The JCL is simple:

```
//TESTTMS EXEC PGM=FATTMTST,PARM=volser
//STEPLIB   DD   DISP=SHR,DSN=fatsfatr.load.library
//SYSPRINT  DD   SYSOUT=*
```

The parameter (PARM=volser) is a tape volume serial. FATTMTST will use the tape management interface to query that volume and will print out the data set name on the volume. If the tape has multiple files on it, all tape files are listed. If the tape is part of a multi-volume tape set, all volumes in the set (and all data sets on the set) are listed. If the volume serial is not recorded in your tape management database, you will get a return code of 12. If you get other return codes, abends, or unexpected results, please contact Innovation for assistance.

Proceed to the next section for product customization.

90.4 CUSTOMIZING FATS, FATAR, AND FATSCOPY

There are no customizable default options in FATS and FATAR. All options for these two products are set via control statement operands. If you have only these two products, **you may skip to section 90.4.2.**

90.4.1 DISPLAYING AND MODIFYING FATSCOPY DEFAULTS

The default values of many FATSCOPY's COPY/SIM keywords are stored in the FATSCOPY Global Options Table. The original values of these defaults are shown as "delivered defaults" in the keyword descriptions in Section 32.2. Each FATSCOPY job reads this table to determine the current default values for each keyword.

The current default values can be displayed and changed using the FATZAPOP program that is provided with FATSCOPY. The keywords for which the default values can be changed are listed in Section 32.2 with asterisks (*), and are also listed below. FATZAPOP displays and/or modifies the FATSCOPY Global Options Table.

You must run FATZAPOP at least one time to define your tape management system to FATSCOPY, as described in Section 90.3.

FATZAPOP JCL

To execute FATZAPOP as a batch job , use the following JCL:

```
//stepname  EXEC  PGM=FATZAPOP,REGION=0M
//STEPLIB   DD    DISP=SHR,DSN=fatsfatr.load.library
//SYSLIB    DD    DISP=SHR,DSN=fatsfatr.load.library
//SYSPRINT  DD    SYSOUT=*
//SYSABEND  DD    SYSOUT=*
//SYSIN     DD    *
 FATZAPOP command statement(s)
/*
```

FATZAPOP will write messages and a report to SYSPRINT.

COPY OPTIONS FROM A PREVIOUS VERSION (4.9.26 or later)

When you upgrade to a new release of FATSCOPY, or from a trial to production version of the same release of FATSCOPY, you will usually want to copy all FATSCOPY default options from the version that you are currently using. FATZAPOP's COPY command statement will copy all defaults from the library whose DD statement is the operand of the FROMDD= keyword:

```
//stepname  EXEC  PGM=FATZAPOP,REGION=0M
//STEPLIB   DD    DISP=SHR,DSN=new.fatsfatr.load.library
//SYSLIB    DD    DISP=SHR,DSN=new.fatsfatr.load.library
//OLDLIB    DD    DISP=SHR,DSN=old.fatsfatr.load.library
//SYSPRINT  DD    SYSOUT=*
//SYSABEND  DD    SYSOUT=*
//SYSIN     DD    *
 COPY   ALL,FROMDD=OLDLIB
/*
```

You can copy default options from releases 4.9.26 or later. Earlier versions of FATSCOPY did not have an Option Table to copy values from.

FATZAPOP COMMAND STATEMENTS

The following command statements can be used with the FATZAPOP program:

- **COPY ALL,FROMDD=oldlibrarydd** – Copy default values from another version of FATSCOPY. This is described in detail above.
- **ZAP keyword1=value1,keyword2=value2,…** – Changes the default value of each **option** to the new **value**. Multiple ZAP command statements may be specified. In most cases, except at noted below, the allowed **keyword**s and **value**s for ZAP are the same as the corresponding keywords and values for the FATSCOPY COPY/SIM statement.
- **PRINT** – Display the current values of all options in the FATSCOPY Global Options Table. Values that have been changed from their delivered defaults (the values shown in Section 32.2) will be marked with asterisks.
- **AUDIT** – Display the current values of the options in the FATSCOPY Global Options Table that have been changed from their delivered defaults. Options that have not had their values changed will not be displayed.
- **RESET** – Resets the values of all options in the FATSCOPY Global Options Table to their delivered default values. The resulting defaults will all match the values shown in this manual, unless RESET is followed by ZAP statements.

  Caution: RESET clears the tape management interface information required by FATSCOPY. **Be sure to follow RESET with a ZAP command statement with the TMSIN= parameter** so that FATSCOPY will recognize your tape management system. If you are using TLMS, a ZAP command statement for the VMFDSN= parameter is also required, as described in Section 90.3.
- **HELP** – Displays information in SYSPRINT about the FATZAPOP program options. The HELP command formats are:

  HELP – show a brief description and list of commands

  HELP SUB(command) – show a detailed subject listing for the specified command (COPY, ZAP, PRINT, RESET, or HELP)

  HELP ALL – display help information for all FATZAPOP commands.

SETTING DEFAULTS WITH THE ZAP STATEMENT

The following FATZAPOP keywords are used with the ZAP command statement to save FATSCOPY defaults in the Global Options Table:

ABEND=	EXPDTGROUP=	MAXFILE=	RMMFAIL=
ABRARC=	HSMBAKMASK=	MAXTERR=	SELTERR=
ABRINDEX=	HSMMIGMASK=	MAXTOTFILE=	SORT=
ADDXDAYS=	HSMML2=	MULTIFILE=	SUPULAB=
AUDITLOG=	LASTFILE=	MULTIVOL=	TMSDATA=
BUFNO=	LINECNT=	NOTCATRC=	TMSIN=
CAT=	LOGERROR=	OFFSITE=	TMSINPUT=
DUMP=	MAJERR=	OPCAP=	VIRTTYPE=
ECHO=	MAXERR=	RETRY=	VMFDSN=

ZAP keyword=value is used to set FATSCOPY default values. The ZAP command takes the same keywords as the corresponding FATSCOPY keywords, and the allowed **value**s are those listed in Section 32.2 for each keyword, except for the following:

- To set “ABEND” as the FATSCOPY default, use ZAP ABEND=U0888. Or, to set “RETCODE” as the FATSCOPY default, use ZAP ABEND=RETCODE. (You cannot use ZAP RETCODE.)
- To set “SUPULAB” as the FATSCOPY default, use ZAP SUPULAB=YES. Or, to set “NOSUPULAB” as the FATSCOPY default, use ZAP SUPULAB=NO. (You cannot use ZAP NOSUPULAB.)

To set **value** to a blank, code ‘ ‘ (single quote, blank, single quote) after the equals sign. Example: ZAP HSMMIGMASK= ‘ ‘

Use ZAP VMFDSN=tlms.vmf.name only when you have TLMS tape management, as described in Section 90.3.

You should carefully consider the implications of changing the defaults for CAT, TMSINPUT, and SORT. **INNOVATION does not recommend changing the defaults for these three parameters.** Changing these defaults may result in unexpected results for the cataloging (CAT) and retention (TMSINPUT) of data sets, and in the number of tape mounts required (SORT).

FATZAPOP EXAMPLE

The following example sets the tape management interface to RMM, and then changes the default values for MULTIFILE, ECHO, and ABRARC. It also uses PRINT to display the new default values for all the options in the Global Options Table.

```
//ZAPOP     EXEC  PGM=FATZAPOP,REGION=0M
//STEPLIB   DD    DISP=SHR,DSN=fatscopy.library.name
//SYSPRINT  DD    SYSOUT=*
//SYSLIB    DD    DISP=SHR,DSN=fatscopy.library.name
//SYSIN     DD    *
 ZAP TMSIN=RMM
 ZAP MULTIFILE=YES,ECHO=YES
 ZAP ABRARC=YES
 PRINT
/*
```

90.4.2 SECURITY

SECURITY

By default, FATAR and FATSCOPY will do security checks on input data sets. It will verify that users have READ authority to each input data set in the DATASET class. This prevents users from using FATAR to access data to which they are not authorized, especially on multi-file tapes, and prevents FATSCOPY from selecting tapes for which the user does not have at least READ authority. More details are found in Sections 21.6 and 30.1. If you do not want to do this data set security check, you have two options:

1. You can apply FATAR/FATSCOPY custom zap C-49.0002 to disable the security check (contact Innovation for assistance); or
2. You can define security resource FATAR.DATASET.SECBYPAS (for FATAR) or FATSCOPY.DATASET.SECBYPAS (for FATSCOPY) in class FACILITY and give all users READ authority to it.

By default, all users can use all functions and options of FATS, FATAR, and FATSCOPY. If you want to restrict certain users from using certain functions, you must apply the FATS/FATAR/FATSCOPY custom zap C-49.0001 and define FACILITY class resources in your security system to control each function, as described in Sections 11.8 (FATS Security), 21.6 (FATAR Security), and 30.1 (FATSCOPY Security).

90.4.3 OPTIONAL MODULES AND TABLES

OPTIONAL MODULES AND TABLES

ICL (Installation Control Library) contains three source modules that you may wish to modify and assemble.

Member **OPENEXIT** is a sample FATAR Open exit. Use of the Open exit is documented in Section 21.4 and is controlled by operand EXIT= documented in Section 23.2.5.

Member **FATCHTAB** is a sample FATAR printer translate table. Use of the translate table is documented under the CHARTAB= operand in Section 23.2.4.

Member **ASCIITRS** is a sample FATAR ASCII translate table, containing a table for translation of ASCII to EBCDIC and another for EBCDIC to ASCII. Use of the translate table is documented under the ASCII= operand in Section 23.2.4.

At this point, installation of the batch function of FATS, FATAR, and FATSCOPY is complete. If you wish to install the ISPF panels, continue to the next section.

90.5 INSTALLING THE ISPF DIALOG

The FATS/FATAR ISPF dialog can be invoked at any time, from any TSO userid that is authorized to read the FATS/FATAR dialog libraries, by going to ISPF option 6 (TSO COMMAND) and entering:

```
EXEC 'fats.clist.library(FATALLOC)'
```

Since this can be awkward, two convenient alternatives are available:

ADDING FATS/FATAR TO A MENU

You may add FATS/FATAR as an option on the ISPF main menu or any menu of your choice. In the FATS/FATAR panel data set, there are two example panels showing how to add FATS/FATAR to the ISPF main menu:

ISR@V3X - for ISPF V3.x **ISR@V4X** - for ISPF V4.x

You can make similar modifications to your ISPF main menu (panel ISR@PRIM) or to any other ISPF menu.

ADDING FATS/FATAR AS A COMMAND

You can add a FATS/FATAR command to the ISPF command table (ISPCMDS) by going to ISPF option 6 (TSO COMMAND) and entering:

```
EXEC 'fats.clist.library(FATCMDS)'
```

The modified command table will be stored in the first library in the ISPTLIB concatenation for this TSO userid. If the user has a private table library, it will normally be the first one in that concatenation, so the updated command table will be available only to this user. If the user does not have a private library, but has update authority to the first public library in ISPTLIB, it will be updated, and will be available to any userid using that library. However, you will have an option to specify a different table library before it is actually stored. If you do choose a different table library, that library will have to be in the ISPTLIB concatenation, prior to any other library containing a table called ISPCMDS. The FATS/FATAR dialog can now be entered from almost any ISPF panel by entering "FATS" on the command line.

CHANGING THE DIALOG LIBRARY NAMES

During the installation with the FATLOAD program, the names of the FATS/FATAR dialog libraries are automatically changed in the CLIST library to the names you specified on the installation screens. If you later change those names, or if you manually loaded the dialog libraries, you can correct the dialog to use the proper names by editing member FATALLOC in the FATS/FATAR CLIST library. Change the appropriate CLIST lines to specify the new data set name(s):

CLIST	FATS/FATAR ISPF Dialog CLIST Library
LLIB	FATS/FATAR Load Module Library
PLIB	FATS/FATAR ISPF Dialog Panel Library
MLIB	FATS/FATAR ISPF Dialog Message Library
SLIB	FATS/FATAR ISPF Dialog Skeleton Library
TLIB	FATS/FATAR ISPF Dialog Table Library

Do not modify the FATSFATR CLIST for this purpose. The table library must be allocated before FATSFATR is invoked.

AUTHORIZING PROGRAMS UNDER TSO

If you want to use the FATSCOPY panels to do foreground simulations, you need to modify member IKJTSOxx in SYS1.PARMLIB. An entry needs to be added to both the AUTHTSF and AUTHPGM tables for program FATSCOPY.

DEFINE JCL PARAMETERS

Each ISPF panel user must define the JCL parameters to be used for FATSCOPY jobs in their ISPF profile. From the Main Menu, select Option “J – JCL PARMS”. Enter any JOB statement parameters required by your installation for running FATSCOPY batch jobs. If FATSCOPY is in the system link list, no Load Library is required; if not, provide the name of the FATS/FATAR/FATSCOPY load library. You can also change the parameter for SYSOUT= statements. Enter END to save these values to the ISPF profile for your userid.

USING THE DIALOG TO ACTIVATE THE TAPE MANAGEMENT INTERFACE

If you are using FATAR or FATSCOPY, the interface for the tape management system you are using must be activated (usually, by the system administrator or system programmer) before any FATAR or FATSCOPY jobs are run. If a FATZAPOP batch job has already been run to activate it, you can skip this step.

Select Option “S - SETOPT” from the Main Menu, then select “2 – Miscellaneous Options” on the following panel (Set FATSCOPY Global Options Primary Menu), and enter the name of your tape management system (RMM, CA1, or TLMS) in the TMSIN section of the panel.

If you are using TLMS tape management, enter F3 to return to the Set FATSCOPY Global Options Primary Menu, then “4 – Data Set Names and Masks”, and then enter the name of your TLMS Volume Master File in the VMFDSN field.

Enter F3 to return to the Set FATSCOPY Global Options Primary Menu. Be sure the correct library is entered in the FATSCOPY LIBRARY field, and enter SAVE on the command line to save your tape management values.

At this point, installation of the ISPF panels is complete.

93.0 ISPF INTERFACE INTRODUCTION

FATS/FATAR/FATSCOPY ISPF panels are available to perform many of the FATS, FATAR, and FATSCOPY functions such as labeling, certifying and copying tapes. These panels create JCL and control statements as job streams that can be submitted, edited before submission, or saved for reuse.

The panels will simplify the use of FATS/FATAR/FATSCOPY for common tasks. Extensive validation of the input is done to eliminate errors at execution time. More complicated tasks such as dropping blocks or replacing data on a tape will be easier to perform.

Options for installing and invoking the FATS/FATAR/FATSCOPY dialogs are shown in Section 90.4.

Every panel has detailed HELP information that can be displayed by pressing the HELP key (usually PF1). If your level of ISPF supports it, field level help is available by placing the cursor in the input field in question and pressing HELP. The descriptions of the fields and functions on the panels in the following sections are brief because the detailed, current descriptions are in the HELP.

The actual ISPF panels may vary slightly from the samples in this manual.

MAIN MENU

```
V4.9.xx ----------------------------  MAIN  MENU  ----------------------------
OPTION ===>

  1. FATS      - Fast Analysis of Tape Surfaces

  2. FATAR     - Fast Analysis of Tape, Tape Recovery and Tape Copy

  3. FATSCOPY  - Automatic Stacking and Migration of high-capacity tapes

  J. JCL PARMS - Specify JCL and SYSOUT defaults for submitted jobs

  S. SETOPT    - Set Installation Options in the FATSCOPY Global Options Table
```

FATS/FATAR/FATSCOPY Main Menu

This is the FATS/FATAR/FATSCOPY Main Menu. **The remainder of this section of the manual covers only the FATS/FATAR panels. The FATSCOPY panels are covered in Section 34.0.**

Two of the selections on this menu are used to configure FATS/FATAR.

- **SETOPT** – This selection is used by the system administrator. **It must be used at least once before any FATAR jobs are run if you want to use the tape management interface.** This is described on page 93-2. However, you can skip this step if you have already set the tape management information by running a FATZAPOP batch job during product installation.

- **JCL PARMS** – Each user needs to select this option the first time they use the FATS/FATAR panels to set JCL defaults in their ISPF user profile. This is described on page 93-3.

If you've already used the JCL PARMS panel, you can select Option 1 to proceed directly to the FATS Selection Menu or Option 2 to proceed to the FATAR Selection Menu.

ACTIVATING THE TAPE MANAGE-MENT INTERFACE

```
V4.9.xx     FATSCOPY INSTALLATION -- SET FATSCOPY GLOBAL OPTIONS PRIMARY MENU
OPTION ==> 


         1 Global Selection Options

         2 Miscellaneous Options

         3 Copy Options

         4 Data Set Names and Masks

  SAVE - Save option changes            COPY - Copy options from a prior level
CANCEL - Exit without saving changes   AUDIT - Display user changed options
                                       RESET - Re-Initialize all options


FATSCOPY LIBRARY DATA SET:
         Data Set Name ===> FATSYS.FAT49XX.LOAD                          <
```

FATSCOPY Global Options Primary Menu

Entering SETOPT on the Main Menu displays this panel. Most of the selections on this panel are used only by FATSCOPY customers, but FATS/FATAR customers also use this panel to activate the tape management interface. Enter "2" to proceed to the following panel:

```
V4.9.xx ------------------- FATSCOPY / MISCELLANEOUS OPTIONS ------------------
COMMAND ===> 


TMSIN      - Active Tape Management System................... RMM   (RMM,CA1,TLMS)
LINECNT    - Max number of print lines per page.............. 60      (1,32767)
MAXFILE    - Max files per output ( 0 equals stack until full).. _____ (1-65535)
ABEND      - Abnormal termination Option ................ ABEND   (ABEND,Retcode)
ECHO       - Echo current options to Sysprint................... NO      (Yes,No)
```

Menu Option 2: Miscellaneous Options

Enter the name of your tape management system and press END (PF3), which returns you to the Global Options Primary Menu. On that panel, enter SAVE to save the tape management information.

JCL INSTALLATION PANEL

```
V4.9.xx ------------------ FATS/FATAR JCL INSTALLATION PANEL ------------------
COMMAND ===>
       Enter END to save JCL to profile           Enter CANcel to Cancel

 Job Statements:
   > //MET0     JOB (ACCOUNT),'NAME',REGION=0M
   > //*
   > //*

  Job Id Suffix: YES Yes to append a character to your Job Name, else enter No

 Load Libraries: FATSYS.T49XX.LOAD                                        <
               : ____________________________________________             <
               : ____________________________________________             <

 Sysout Dataset: *                                                        <

                 ---------- FATS & FATAR Options ------------
 Volser Default: 99999 enter default Volume Serial for input DD Statement.
                       - (6th digit is system generated)
    Save DSName: NO  Yes to save Dataset Names in your Profile, else enter No
```

Option J – JCL Installation Panel

The JCL Installation panel is used to set JCL defaults in your ISPF user profile. This needs to be done by each user. Enter any JOB statement parameters required by your installation for running FATS/FATAR batch jobs. If FATS/FATAR is in the system link list, no Load Library is required; if not, provide the name of the FATS/FATAR load library so that the correct STEPLIB DD statements can be generated. You can also change the parameter for SYSOUT= statements.

Enter END to save these values to the ISPF profile for your userid and return to the Main Menu.

You may now select Option 1 from the Main Menu to proceed to the FATS Selection Menu or Option 2 to proceed to the FATAR Selection Menu (if you are licensed for FATAR).

TAPE MANAGEMENT SUPPORT

If you have used the SETOPT option or run a batch job to identify the tape management program active on your system (see Section 90.3, or the Option 2 panel above), option Q will appear on the FATAR Selection Menu. (Option Q is not available if you are licensed for only FATS.) Option Q allows you to query your tape management database through facilities provided by FATS/FATAR.

For a given tape volume serial, option Q will display all the data sets recorded as being on that tape. If the tape is part of a multi-volume tape set, all volumes and all data sets in the set are displayed, also showing which volumes each data set resides on.

When the tape management interface is installed, you may also use a Q on the command line of many of the panels for the various FATAR functions (a prompt will appear on the panel when the function is available). This allows you to query your tape management system and automatically copy data set name, label type, and file number information into the appropriate fields for the input or output tapes.

FATS/FATAR SETUP/PROFILE PANELS

To set up the FATS and FATAR profiles, you need to select Option '1' for FATS or Option '2' for FATAR from the Main Menu. The panel below will be displayed when Option '1' is selected from the Main Menu.

```
V4.9.xx ------------------------- FATS SELECTION MENU -------------------------
OPTION ===>

            1. CERTIFY new tapes

            2. CERTIFY existing labeled tapes whose data sets have expired

            3. LABEL   new tapes

            4. ERASE   old data on tapes for security

            5. VERIFY  whether tapes are still readable


            P. FATS    Application and JCL defaults
```

Option 1: FATS Selection Menu

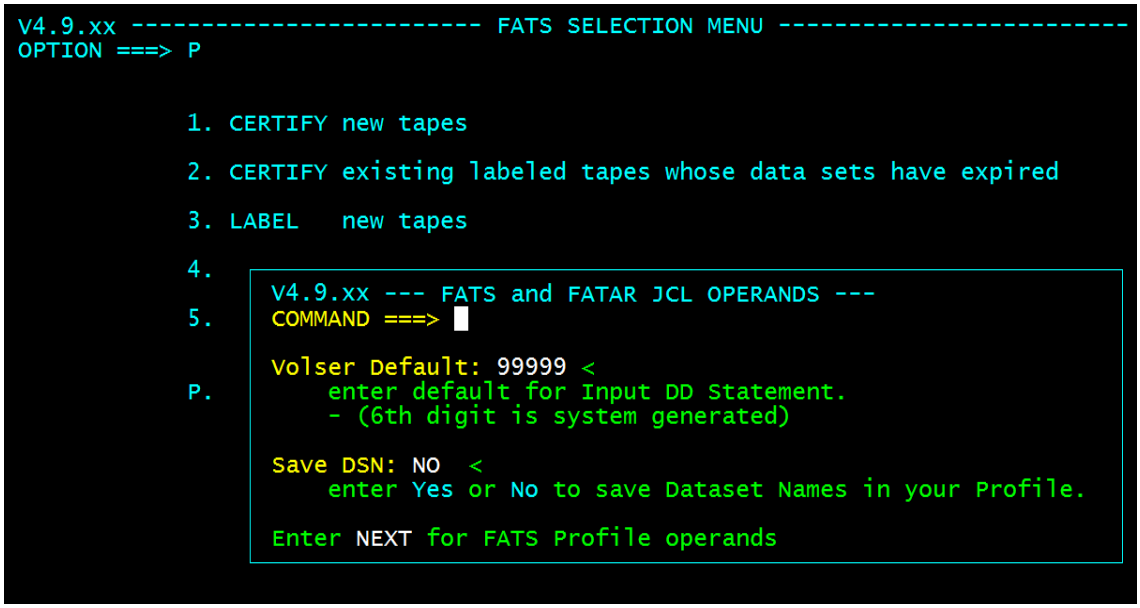
The first time you use this dialog, you will need to set up some FATS/FATAR defaults for your userid using option 'P' from the Selection Menu. The panels set defaults that will be used during the generation of job streams by other options on the menu. There are 3 kinds of defaults:

- Volser and DSN options, for generation of JOB statement and other JCL parameters.
- FATS and FATAR "Profile" options, giving values for operands, such as lines/page, that **cannot** be overridden on other panels. The defaults provided by Innovation for these values will rarely need to be overridden, but if necessary you can set them here to the values you need.
- FATS and FATAR "Control" options, giving values for operands that **can** be overridden on other panels. However, setting these to the values you most commonly use will save time and typing later.

You can return to these panels at any time to change your saved options. These options are saved in your ISPF PROFILE data set, so they affect only your own TSO userid.

On the FATS and FATAR function panels, an option will always appear asking if you want to see more FATS or FATAR operands. If you reply YES, you will see a panel similar to the "Control" panels shown below. If you change values on those panels, they are **not** saved in your profile and will need to be reentered every time you need them.

JOB PROFILE When you enter Option 'P' from either the FATS or FATAR Selection Menu, the following pop-up panel will be displayed.


V4.9.xx ------------------------- FATS SELECTION MENU -------------------------
OPTION ===> P
1. CERTIFY new tapes
2. CERTIFY existing labeled tapes whose data sets have expired
3. LABEL new tapes
4.
5.
P.
V4.9.xx --- FATS and FATAR JCL OPERANDS ---
COMMAND ===>
Volser Default: 99999 <
enter default for Input DD Statement.
- (6th digit is system generated)
Save DSN: NO <
enter Yes or No to save Dataset Names in your Profile.
Enter NEXT for FATS Profile operands

Option P: FATS/FATAR Profile: JCL Operands

This panel allows you to enter a default volume serial to be used by FATS/FATAR for creating input DD statements, and to decide if FATS/FATAR should save the data set names used on the panels for TAPEIN and TAPEOUT DD statements.

When done, enter NEXT to advance to the FATS profile operands panel or the END key (usually PF3 or PF15) to save the profile.

FATS DEFAULT PROFILE OPERANDS

```
V4.9.xx ------------------- FATS DEFAULT "PROFILE" OPERANDS -------------------
COMMAND ===>

LINECNT   - Number of lines per page to be printed on FATS reports: 60
MAXCLEAN  - Clean tape drive after ___ tapes are processed on drive
MAXDETCNT - Number of detail lines to be printed before double spacing: 4
METRIC    - List tape lengths/error positions in metric units: NO  (yes/no)
MODIFY    - Respond to operator's MODIFY(F) and STOP(P) commands : YES (yes/no)
NEWPAGE   - Skip to new page when a new volume is processed  : NO  (yes/no)
RETCODE   - Terminate with return code 12 due to major error : YES (yes/no)
REWIND    - Rewind tapes to load point at start of operation : YES (yes/no)
WTO       - Report result of each operation on system console: NO  (yes/no)
WTOR      - Issue WTOR message for operator control of processing: NO  (yes/no)

    Enter NEXT for FATS Control Operands          Enter PREV for JCL Profile
```

FATS Default Profile Operands

This panel modifies the default options that will be used for many FATS operations. These operands are **not** overrideable from the function panels. Most users will require no changes except for those users who would prefer the lengths be shown in metric units (the METRIC option). When done, enter NEXT to see the next panel, PREV to return to the JCL profile, or the END key (usually PF3 or PF15) to save the profile.

FATS DEFAULT CONTROL OPERANDS

```
V4.9.xx ------------------- FATS DEFAULT "CONTROL" OPERANDS -----------------
COMMAND ===>

        The entries in these operands will be retained in your profile.

BLKSIZE  - Size of block written to new tape during Certification: _____
DDLABEL  - Label tape with volume serial in TAPE1 DD statement: NO  (yes/no)
ERRCLEAN - Clean tape drive after 32767 temporary or permanent errors
LABEL    - Label tape with volume serial number: _____
MAXERR   - Stop processing after 200   permanent errors encountered
MAXCERR  - Stop processing after 200   inches of contiguous permanent errors
MAXTERR  - Stop processing after 200   temporary data checks are detected
MODE     - Recording Mode for Drive: __    (refer to tutorial/manual)
NUMWRITES- Nbr of writes that FATS will do in one I/O when Certifying: 20
OPERATOR - Invoke OPERator/NOOPerator option: ____  (oper/noop)
OWNERID  - Contents of owner ID field in output volume label: _________
RETRY(H) - Number of retries before error is considered permanent: 10
RETRY(I) - Number of retries before error is printed in detail report: 1
STOP     - Stop/Issue FATSW05 message for each permanent error: NO  (yes/no)
STOPNUM  - Invoke STOP option after _ permanent errors have been encountered
THRESHOLD- Stop printing error messages after 1000  permanent/temporary errors

    Enter PREV for FATS Control operands
```

FATS Default Control Operands

This panel modifies the default options that will be used for many FATS operations. These operands **are** overrideable from the function panels. When done, enter PREV to return to the FATS Control Operands panel or the END key (usually PF3 or PF15) to save the profile.

FATAR DEFAULT PROFILE OPERANDS

The FATAR Default Profile Operands Panel will be displayed when you enter option '2' from the Main Menu, select Option 'P from the FATAR Selection Menu, and enter 'NEXT' from the FATS and FATAR JCL Profile Menu.

```
V4.9.xx ------------------- FATAR DEFAULT "PROFILE" OPERANDS ------------------
COMMAND ===>


BYTEFACTOR- Specify divisor to calculate byte count values for reports: 1000
EXITNAME  - Name of load module to be executed at open time: ________
LINECNT   - Number of lines per page to be printed on FATAR reports: 60
MAJERR    - Stop processing after 1     major errors are encountered
MAXERR    - Stop processing after 20    permanent data checks are detected
MAXTERR   - Stop processing after 100   temporary data checks are detected
METRIC    - List tape lengths/error positions in metric units    : NO  (yes/no)
MSGLEVEL  - Print contents of modified blocks?                   : YES (yes/no)
RETCODE   - Terminate with return code 12 due to major error     : YES (yes/no)
ROR       - Attempt "Read-Opposite-Recovery" on round reels      : YES (yes/no)
SETNRC    - Terminate with return code 4 or 8 for unusual errors: NO  (yes/no)
WTO       - Report result of each operation on system console    : NO  (yes/no)

    Enter NEXT for FATAR Control Operands          Press PREV for JCL Profile
```

FATAR Default Profile Operands

This panel modifies the default options that will be used for many FATAR operations. These operands are **not** overrideable from the function panels. Most users will require no changes except for those users who would prefer the lengths be shown in metric units (the METRIC option). When done, enter NEXT to see the next panel, PREV to return to the JCL Profile panel, or the END key (usually PF3 or PF15) to save the profile.

FATAR DEFAULT CONTROL OPERANDS

```
V4.9.xx -------------------- FATAR DEFAULT "CONTROL" OPERANDS -----------------
COMMAND ===>

        The entries in these operands will be retained in your profile.

ASCII     - Load module containing translation table for tapeout : ________
CHARTAB   - Load module containing translation table for printing: ________
CHECKTMS  - Issue Tape Mgmt queries agains input file: NO  (yes/no)
CREATEDATE- Copy creation date from input tape labels: NO  (yes/no)
DATAFMT   - Print format of blocks/records: DUMP (char/hex/dump)
EOTBLOCKS - Nbr of blocks to write after EOT for Image copy: ___
EOV       - Check for EOV labels/update block counts : YES (yes/no)
KEEP      - Copy block even on a permanent data check: NO  (yes/no)
LBLPRT    - Print format of input labels: FORMAT (char/hex/dump/format/none)
LENCHK    - Print contents of different block lengths: NO  (yes/no)
MAXRBLK   - Largest current blocksize for reblocking : ______
OUTFSEQ   - File Seq Nbr to be assigned to 1st file written to TAPEOUT: 1
PRTLEN    - Nbr of bytes/characters of input tape data to be printed: 80
PRINTTMS  - Print TMS list for selected Volume Serial: NO  (yes/no)
REBLOCK   - Target blocksize for data reblocking: ______
RETRY     - Number of retries before error will be considered permanent: 40
STOP      - Provide operator control at End-of-File: ___ (yes/no)
USERLABELS- Copy user labels to output tape labels : YES (yes/no)
VIRTUNIT  - Assigned esoteric device name for virtual tape units: ________

     Enter PREV for FATAR Profile Operands
```

FATAR Default Control Operands

This panel modifies the default options that will be used for many FATAR operations. These operands are overrideable from the function panels. When done, hit the END key (usually PF3 or PF15) to save the profile or PREV to return to the previous panel.

93.1 FATS ISPF INTERFACE

FATS SELECTION MENU

V4.9.xx ---------- FATS SELECTION MENU ----------
OPTION ===>
1. CERTIFY new tapes
2. CERTIFY existing labeled tapes whose data sets have expired
3. LABEL new tapes
4. ERASE old data on tapes for security
5. VERIFY whether tapes are still readable
P. FATS Application and JCL defaults

FATS Selection Menu

Options 1 through 5 on the FATS Selection Menu will generate job streams for FATS functions. Option P, described in the section FATS/FATAR SETUP/PROFILE PANELS starting on page 93-4, is used to set up default values used by FATS.

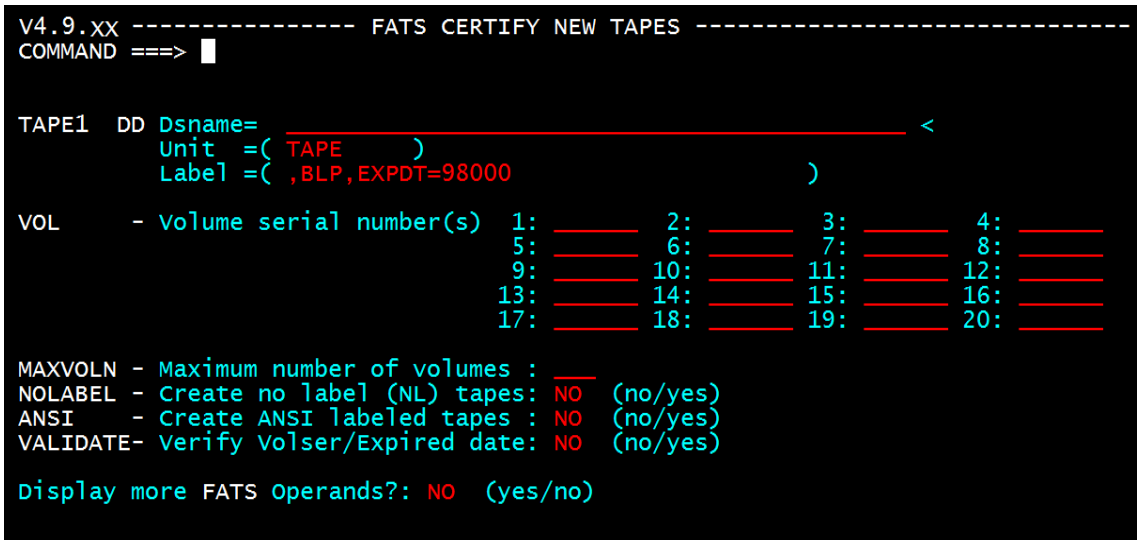
FATS OPERATIONS

Once you enter all necessary options on a given panel, press ENTER to complete generation of the FATS batch job stream. You will be given the option of submitting the job stream immediately or entering ISPF EDIT to view and modify the job stream; you can submit it directly from EDIT, or save it in a JCL library

All the panels contain a data set name, unit name, and label parameter. A unique default data set name may be generated but you may overtype it with another name (this may be required to meet security restrictions); the name must be present to meet JCL requirements, but is usually not actually used by FATS. The unit name should be overridden with a generic name (such as 3490) or esoteric name (such as CART) that will allocate a tape drive of the appropriate type; a specific unit name (such as 3A0) may also be used. The label type presented by the panel is usually appropriate for the function requested but you may overtype it if required.

All panels contain those FATS options that are commonly used with the function on that panel; you can override the defaults displayed as necessary. They also contain a question "Display more FATS Operands?". If you enter YES, the FATS DEFAULT “CONTROL” OPERANDS panel shown in Section 93.0 is displayed, where you can override less commonly used parameters. The overrides are effective for this function only; the values are not saved.

FATS CERTIFY NEW TAPES


V4.9.XX ---------- FATS CERTIFY NEW TAPES ----------
COMMAND ===>
TAPE1 DD Dsname= <
Unit =(TAPE)
Label =(,BLP,EXPDT=98000)
VOL - Volume serial number(s) 1: 2: 3: 4:
5: 6: 7: 8:
9: 10: 11: 12:
13: 14: 15: 16:
17: 18: 19: 20:
MAXVOLN - Maximum number of volumes :
NOLABEL - Create no label (NL) tapes: NO (no/yes)
ANSI - Create ANSI labeled tapes : NO (no/yes)
VALIDATE- Verify Volser/Expired date: NO (no/yes)
Display more FATS Operands?: NO (yes/no)

Option 1: Certify New Tapes

Option 1 (CERTIFY new tapes) on the FATS Selection Menu brings you to this panel, to certify and label new tapes (or relabel existing tapes). Normally it creates IBM standard labeled tapes, but you can create ANSI (AL) tapes or unlabeled tapes by specifying YES to the proper option. Since these are usually virgin (never used) tapes, the BLP option must be used to mount them without trying to read existing labels.

Tape volume serials can be specified in two ways:

- Specify up to 20 specific volume serials.
- Specify the first volume serial after "1:" and specify the number of volumes after "MAXVOLN". A range of sequentially number volumes will be initialized.

FATS CERTIFY EXISTING TAPES

```
V4.9.XX ---------------- FATS CERTIFY EXISTING TAPES -------------------------
COMMAND ===>

TAPE1  DD Dsname=  ____________________________________________ <
          Unit  =( TAPE     )
          Label =( ,SL                                )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN - Maximum number of volumes : ___
NOLABEL - Create no label (NL) tapes: NO  (no/yes)
ANSI    - Create ANSI labeled tapes : NO  (no/yes)
VALIDATE- Verify Volser/Expired Date: NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)
```

Option 2: Certify Existing Tapes

Option 2 (CERTIFY existing labeled tapes) on the FATS Selection Menu brings you to this panel, to certify existing labeled tapes. This might be used to periodically recertify tapes while they are in scratch status. Since these are existing labeled tapes, SL allows z/OS to verify that the proper tapes is mounted and in scratch status before beginning the certification.

If you have a range of tapes to recertify, you can specify a starting serial number and the maximum number of volumes. It is more likely that you will have a list of random serials to process; simply enter up to 20 such serials.

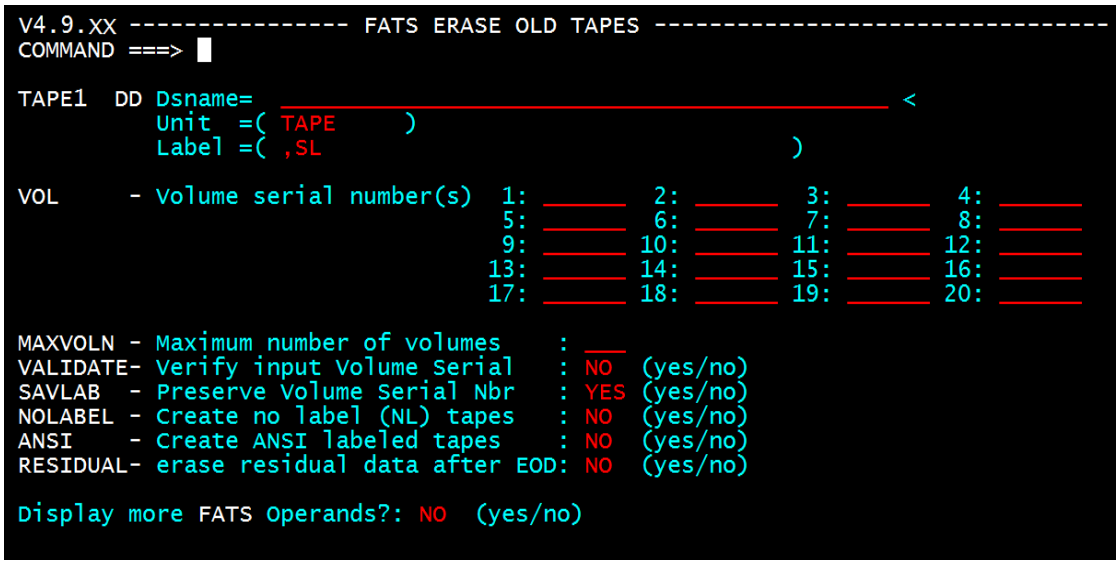
FATS LABELING OF NEW TAPES

```
V4.9.XX ---------------- FATS LABEL NEW TAPES --------------------------------
COMMAND ===>

TAPE1  DD Dsname=  ____________________________________________ <
          Unit  =( TAPE     )
          Label =( ,BLP,EXPDT=98000                   )

VOL     - Volume serial number(s)  1: ______  2: ______  3: ______  4: ______
                                   5: ______  6: ______  7: ______  8: ______
                                   9: ______ 10: ______ 11: ______ 12: ______
                                  13: ______ 14: ______ 15: ______ 16: ______
                                  17: ______ 18: ______ 19: ______ 20: ______

MAXVOLN - Maximum number of volumes : ___
NOLABEL - Create no label (NL) tapes: NO  (no/yes)
ANSI    - Create ANSI labeled tapes : NO  (no/yes)
VALIDATE- Verify Volser/Expired date: NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)
```

Option 3: Label New Tapes

Option 3 (LABEL) on the FATS Selection Menu brings you to this panel, that labels (but does not certify) new tapes or relabels existing tapes. The parameters are the same as for Option 1 (CERTIFY new tapes).

FATS ERASE TAPE

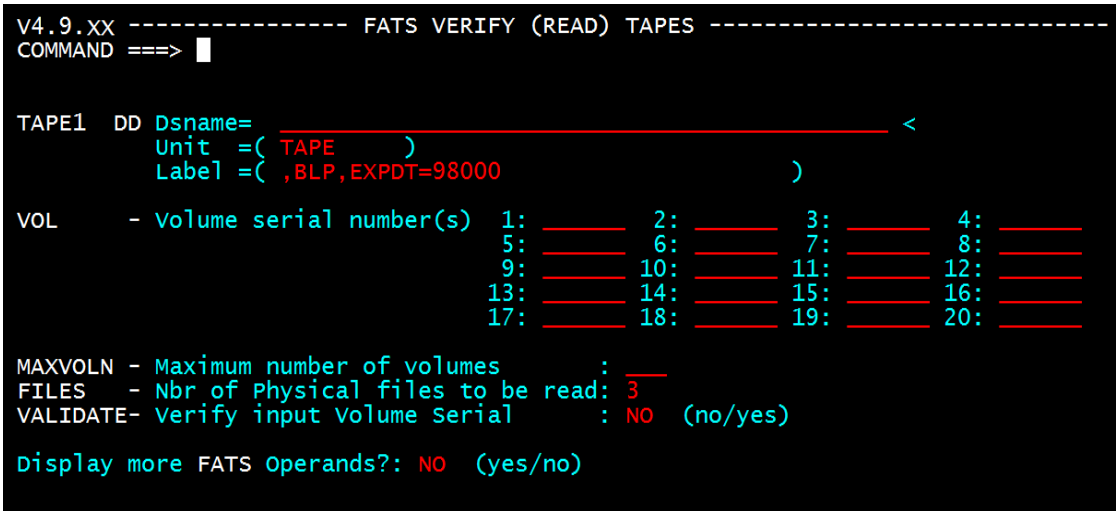

V4.9.XX ---------- FATS ERASE OLD TAPES ----------
COMMAND ===>
TAPE1 DD Dsname=
Unit =(TAPE)
Label =(,SL)
VOL - Volume serial number(s) 1: 2: 3: 4:
5: 6: 7: 8:
9: 10: 11: 12:
13: 14: 15: 16:
17: 18: 19: 20:
MAXVOLN - Maximum number of volumes : NO (yes/no)
VALIDATE- Verify input Volume Serial : NO (yes/no)
SAVLAB - Preserve Volume Serial Nbr : YES (yes/no)
NOLABEL - Create no label (NL) tapes : NO (yes/no)
ANSI - Create ANSI labeled tapes : NO (yes/no)
RESIDUAL- erase residual data after EOD: NO (yes/no)
Display more FATS Operands?: NO (yes/no)

Option 4: Erase Old Tapes

Option 4 (ERASE) on the FATS Selection Menu brings you to this panel, which allows you to erase all data from one or more tapes. This might be used before selling old tape volumes or before writing a volume with data to be sent offsite, to ensure that all previous corporate data is removed. The parameters are the same as for Option 2 (CERTIFY existing labeled tapes), except for the SAVLAB option that allows you to preserve the existing volume serial.

WARNING: This option will erase all data on a tape volume. Use caution when using this option.

FATS VERIFY OF EXISTING DATA ON TAPE


V4.9.XX ---------- FATS VERIFY (READ) TAPES ----------
COMMAND ===>
TAPE1 DD Dsname= <
Unit =(TAPE)
Label =(,BLP,EXPDT=98000)
VOL - Volume serial number(s) 1: 2: 3: 4:
5: 6: 7: 8:
9: 10: 11: 12:
13: 14: 15: 16:
17: 18: 19: 20:
MAXVOLN - Maximum number of volumes :
FILES - Nbr of Physical files to be read: 3
VALIDATE- Verify input Volume Serial : NO (no/yes)
Display more FATS Operands?: NO (yes/no)

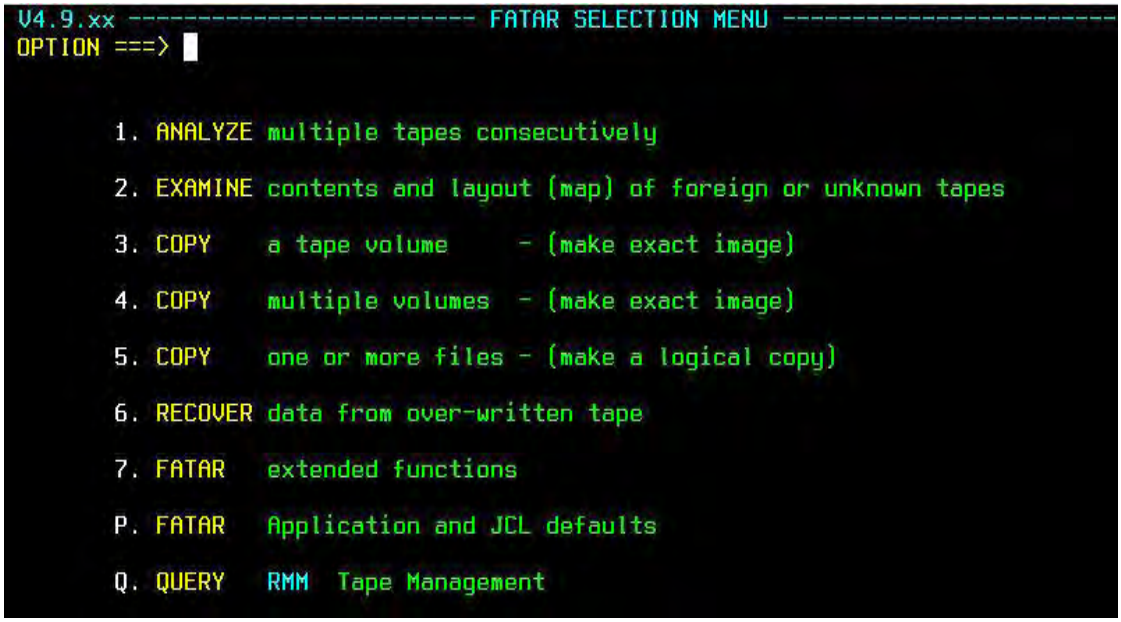
Option 5: Verify Existing Data on Tape

Option 5 (VERIFY) on the FATS Selection Menu brings you to this panel, which uses the FATS READ function to verify the readability of data on the specified tape volumes. The volumes to be processed are specified as described for Option 1 (CERTIFY new tapes). The default of reading 3 physical files will read one logical data file (header file, physical data file, and trailer file) on a labeled tape.

Note: If you are licensed for FATAR, options 1 and 2 from the FATAR Selection Menu provide more information about the content of the tape than option 5. FATAR is the preferred method for verifying tapes.

93.2 FATAR ISPF INTERFACE

FATAR SELECTION MENU


V4.9.xx ---------- FATAR SELECTION MENU ----------
OPTION ===>
1. ANALYZE multiple tapes consecutively
2. EXAMINE contents and layout (map) of foreign or unknown tapes
3. COPY a tape volume - (make exact image)
4. COPY multiple volumes - (make exact image)
5. COPY one or more files - (make a logical copy)
6. RECOVER data from over-written tape
7. FATAR extended functions
P. FATAR Application and JCL defaults
Q. QUERY RMM Tape Management

FATAR Selection Menu

The FATAR Selection Menu will be displayed when you select Option 2 (FATAR Fast Analysis of Tape and Recovery) from the Main Menu.

FATAR OPERATIONS

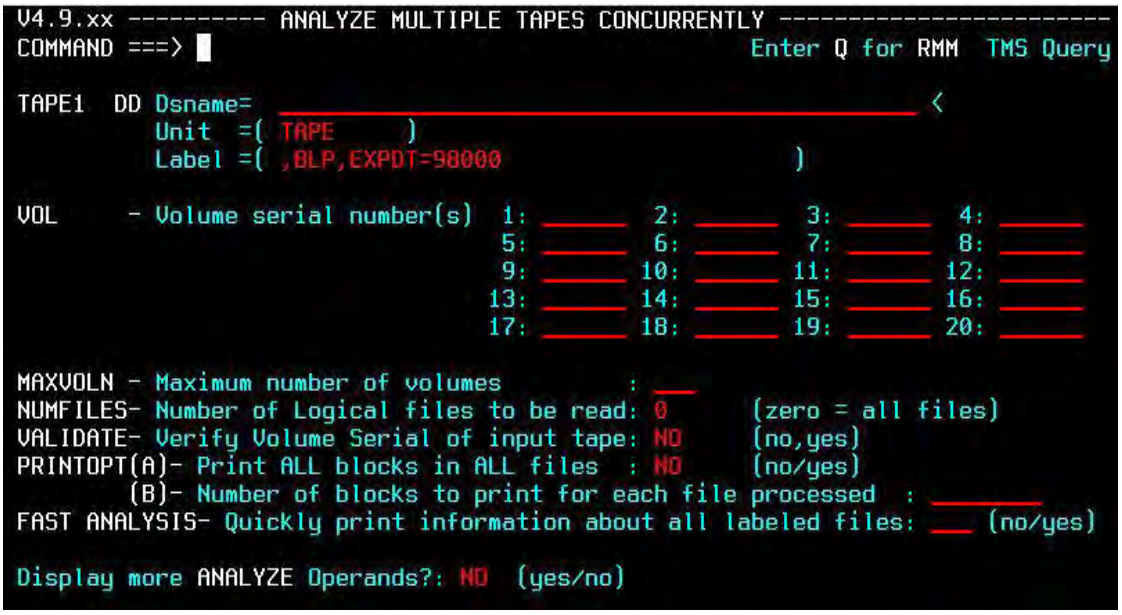
Options 1 through 7 on the FATAR Selection Menu will generate job streams for FATAR operations and will work only if you are licensed for FATAR.

Once you enter all necessary options on a given panel (except for option 7, which is described later), press ENTER to complete generation of the FATAR batch job stream. You will be given the option of submitting the job stream immediately or entering ISPF EDIT to view and modify the job stream; you can submit it directly from EDIT, or save it in a JCL library for future use.

All the panels contain a data set name, unit name, and label parameter. A unique default data set name may be generated but you may overtype it with another name (this may be required to meet security restrictions). If you are reading labeled tapes (LABEL=SL or AL) you must enter the actual data set name recorded on the tape label, but if you are bypassing labels (LABEL=BLP) any valid name may be used. For output tapes, the data set name you specify may be overridden by FATAR. The unit name should be overridden with a generic name (such as 3490) or esoteric name (such as CART) that will allocate a tape drive of the appropriate type; a specific unit name (such as 3A0) may also be used.

All panels contain those FATAR options that are commonly used with the function on that panel; you can override the defaults displayed as necessary. They also contain a question "Display more FATAR Operands?". If you enter YES, the FATAR DEFAULT “CONTROL” OPERANDS panel shown in Section 93.0 is displayed, where you can override less commonly used parameters. The overrides are effective for this function only; the values are not saved.

ANALYZE MULTIPLE TAPES CONCURRENTLY


V4.9.xx ---------- ANALYZE MULTIPLE TAPES CONCURRENTLY ----------
COMMAND ===>
Enter Q for RMM TMS Query
TAPE1 DD Dsname=
Unit =(TAPE)
Label =(,BLP,EXPDT=98000)
VOL - Volume serial number(s) 1: 2: 3: 4:
5: 6: 7: 8:
9: 10: 11: 12:
13: 14: 15: 16:
17: 18: 19: 20:
MAXVOLN - Maximum number of volumes :
NUMFILES- Number of Logical files to be read: 0 (zero = all files)
VALIDATE- Verify Volume Serial of input tape: NO (no,yes)
PRINTOPT(A)- Print ALL blocks in ALL files : NO (no/yes)
(B)- Number of blocks to print for each file processed :
FAST ANALYSIS- Quickly print information about all labeled files: (no/yes)
Display more ANALYZE Operands?: NO (yes/no)

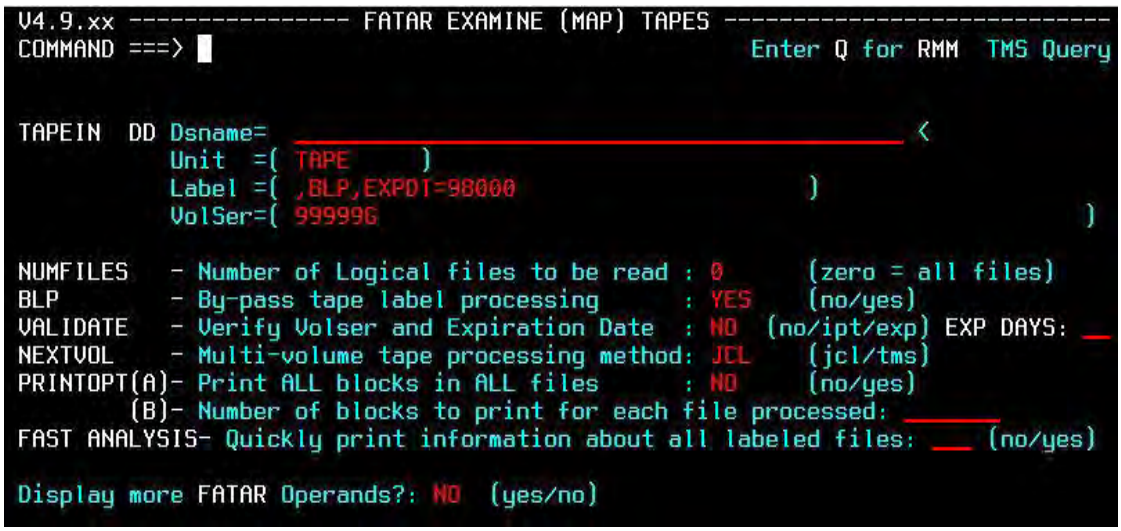
Option 1: Analyze Multiple Tapes Concurrently

Option 1 on the FATAR Selection Menu brings you to this panel, which uses the FATS ANALYZE function to invoke FATAR to analyze the specified tape volumes concurrently. The volumes to be processed are specified as described for Option 1 (CERTIFY new tapes) on the FATS Selection Menu. The PRINTOPT options allow you to print data from the tapes, or to print only the labels from the first file on the tape (to quickly identify the contents of tapes).

Option 1 will function only if you are licensed for FATAR.

Note: Option 2 (EXAMINE) can be used for the same function. Option 2 must be used if you want to analyze a multi-volume tape set and get a combined report on the entire set. Option 1 treats each tape volume as an independent tape and produces a separate report for each. For a single tape volume, use option 2.

EXAMINE CONTENTS AND LAYOUT (MAP) OF TAPES

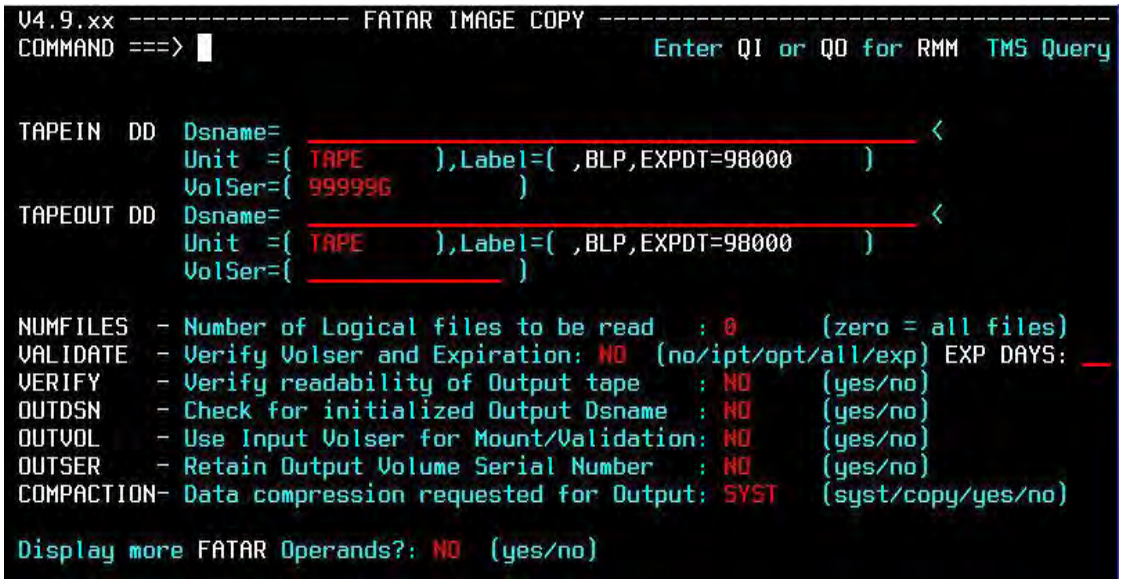

V4.9.xx ---------- FATAR EXAMINE (MAP) TAPES ----------
COMMAND ===>
Enter Q for RMM TMS Query
TAPEIN DD Dsname=
Unit =(TAPE)
Label =(,BLP,EXPDT=98000)
VolSer=(999996)
NUMFILES - Number of Logical files to be read : 0 (zero = all files)
BLP - By-pass tape label processing : YES (no/yes)
VALIDATE - Verify Volser and Expiration Date : NO (no/ipt/exp) EXP DAYS:
NEXTVOL - Multi-volume tape processing method: JCL (jcl/tms)
PRINTOPT(A)- Print ALL blocks in ALL files : NO (no/yes)
(B)- Number of blocks to print for each file processed:
FAST ANALYSIS- Quickly print information about all labeled files: (no/yes)
Display more FATAR Operands?: NO (yes/no)

Option 2: Examine Tapes

Option 2 on the FATAR Selection Menu brings you to this panel, which invokes FATAR to analyze a tape. If you use the default of BLP, the data set name and volume serial (volser) are not actually verified against the tape, so any values required to get the proper tape mounted and opened can be used; this allows any tape, including tapes with unknown contents and tapes from outside your installation to be mounted. If you change the label type to SL or AL, the actual data set name and volume serial must be entered. Multiple volume serials may be entered if they are part of a multi-volume tape set. FATAR will report on the actual volume serial, label type, data set names, and contents of every file on the tape. The PRINTOPT options allow you to print data from the tapes, or to print only the labels from the first file on the tape (to quickly identify the contents of tapes).

Note: Option 1, as shown in the previous figure, can be used for the same function. However, when multiple tape serials are entered on Option 1, they are treated as individual tapes, not as part of a tape set, and each tape receives a separate FATAR report. Option 2 requires that multiple serials be part of a multi-volume tape set. Option 2 should be used for a single tape volume.

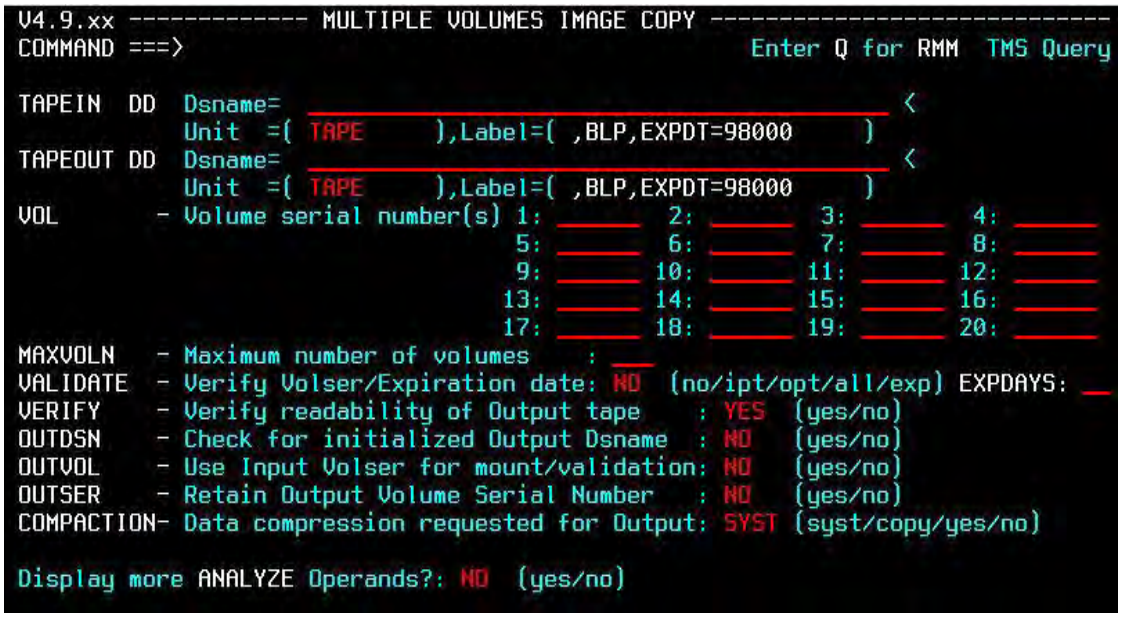
MAKE AN EXACT IMAGE OF A TAPE VOLUME


V4.9.xx ---------- FATAR IMAGE COPY ----------
COMMAND ===>
Enter QI or QO for RMM TMS Query
TAPEIN DD Dsname=
Unit =(TAPE),Label=(,BLP,EXPDT=98000)
VolSer=(99999G)
TAPEOUT DD Dsname=
Unit =(TAPE),Label=(,BLP,EXPDT=98000)
VolSer=( )
NUMFILES - Number of Logical files to be read : 0 (zero = all files)
VALIDATE - Verify Volser and Expiration: NO (no/ipt/opt/all/exp) EXP DAYS:
VERIFY - Verify readability of Output tape : NO (yes/no)
OUTDSN - Check for initialized Output Dsname : NO (yes/no)
OUTVOL - Use Input Volser for Mount/Validation: NO (yes/no)
OUTSER - Retain Output Volume Serial Number : NO (yes/no)
COMPACTION- Data compression requested for Output: SYST (syst/copy/yes/no)
Display more FATAR Operands?: NO (yes/no)

Option 3: Image Copy of a Tape Volume

Option 3 on the FATAR Selection Menu brings you to this panel, which generates a FATAR job to make an image copy (an exact bit-for-bit copy) of a single tape volume. The submitted job **must** be authorized to use BLP (Bypass Label Processing) since this function is required for image copy. Since BLP is used on TAPEIN, any dsname and volser can be specified, whatever is sufficient to mount the proper input tape. However, if INPUT or ALL is specified for VALIDATE, the volume serial of the input tape will be checked against the serial you provided. The dsname of TAPEOUT will not be used, and the TAPEOUT volume serial is whatever you require to get the proper output tape mounted. The output tape will have the actual volume serial of the input tape after the copy is complete, unless you specify YES for OUTSER.

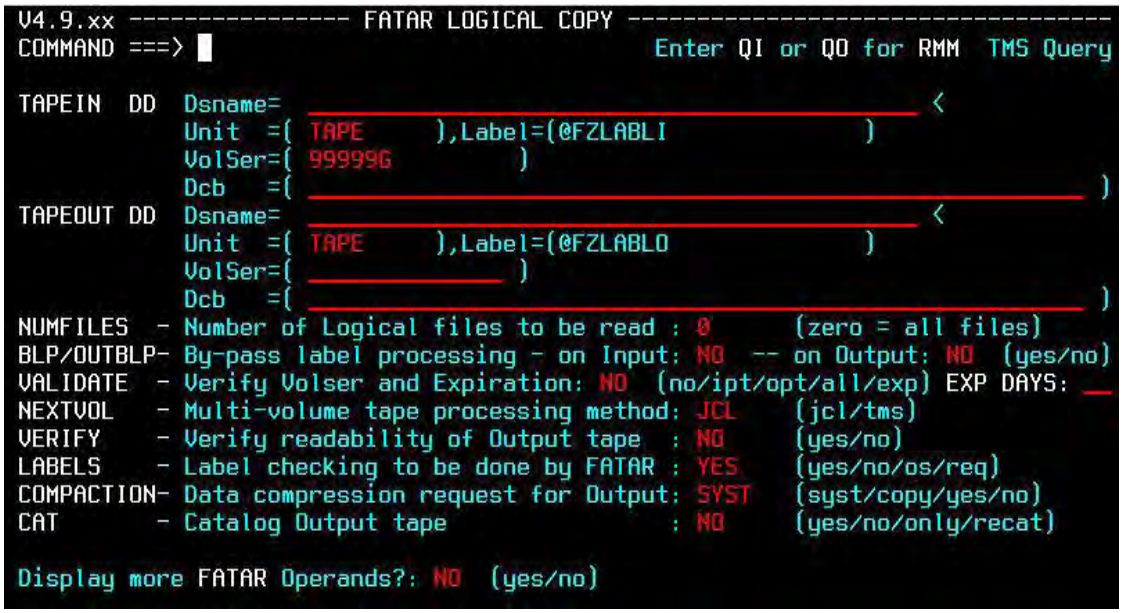
IMAGE COPY OF MULTIPLE TAPES


V4.9.xx ---------- MULTIPLE VOLUMES IMAGE COPY ----------
COMMAND ===>
Enter Q for RMM TMS Query
TAPEIN DD Dsname=
Unit =(TAPE),Label=(,BLP,EXPDT=98000)
TAPEOUT DD Dsname=
Unit =(TAPE),Label=(,BLP,EXPDT=98000)
VOL - Volume serial number(s) 1: 2: 3: 4:
5: 6: 7: 8:
9: 10: 11: 12:
13: 14: 15: 16:
17: 18: 19: 20:
MAXVOLN - Maximum number of volumes :
VALIDATE - Verify Volser/Expiration date: NO (no/ipt/opt/all/exp) EXPDAYS:
VERIFY - Verify readability of Output tape : YES (yes/no)
OUTDSN - Check for initialized Output Dsname : NO (yes/no)
OUTVOL - Use Input Volser for mount/validation: NO (yes/no)
OUTSER - Retain Output Volume Serial Number : NO (yes/no)
COMPACTION- Data compression requested for Output: SYST (syst/copy/yes/no)
Display more ANALYZE Operands?: NO (yes/no)

Option 4: Copy Multiple Volumes

Option 4 on the FATAR Selection Menu brings you to this panel, which uses the ANALYZE option of FATS to invoke FATAR to perform an image copy of multiple input tapes to multiple output tapes. The input tapes might comprise a multi-volume tape set, or might be unrelated; since image copy completely copies one input tape to one output tape, the relation of the input tapes is immaterial. The considerations for the TAPEIN and TAPEOUT data set names are the same as for Option 3 (COPY a tape volume) and the rules for specifying the input volume serials are the same as for Option 1 (CERTIFY new tapes) from the FATS Selection Menu. However, the job will call for scratch (PRIVAT) mounts for each output tape; your operator must be prepared to mount the proper output tape for each input tape requested.

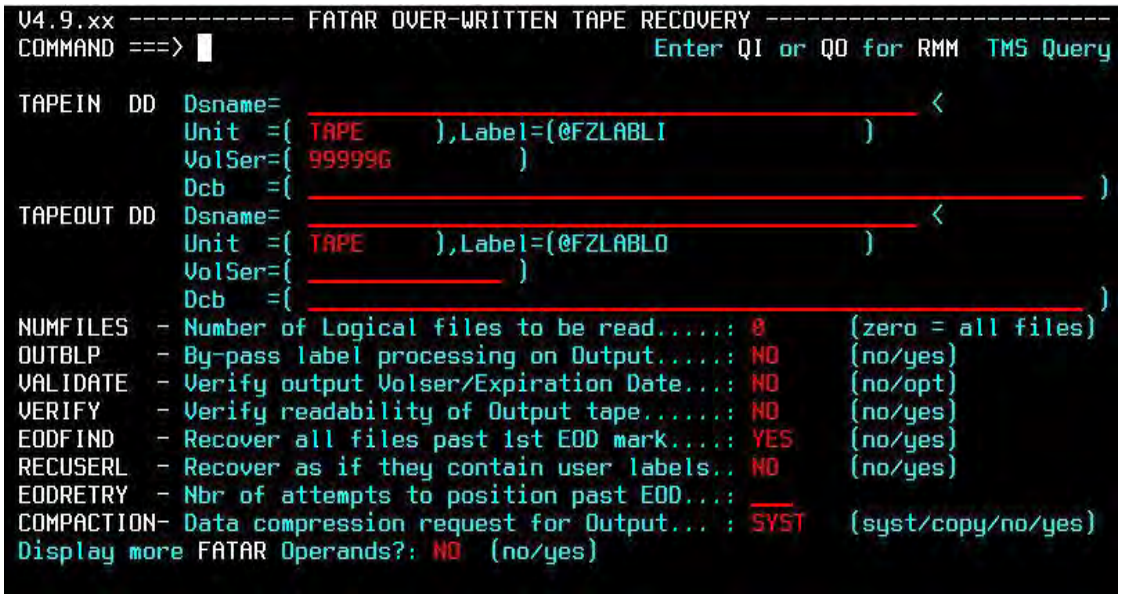
MAKE A LOGICAL COPY OF ONE OR MORE FILES


V4.9.xx ---------- FATAR LOGICAL COPY ----------
COMMAND ===>
Enter QI or QO for RMM TMS Query
TAPEIN DD Dsname=
Unit =(TAPE),Label=(@FZLABLI)
VolSer=(99999G)
Dcb =(
TAPEOUT DD Dsname=
Unit =(TAPE),Label=(@FZLABLO)
VolSer=(
Dcb =(
NUMFILES - Number of Logical files to be read : 0 (zero = all files)
BLP/OUTBLP- By-pass label processing - on Input: NO -- on Output: NO (yes/no)
VALIDATE - Verify Volser and Expiration: NO (no/ipt/opt/all/exp) EXP DAYS:
NEXTVOL - Multi-volume tape processing method: JCL (jcl/tms)
VERIFY - Verify readability of Output tape : NO (yes/no)
LABELS - Label checking to be done by FATAR : YES (yes/no/os/req)
COMPACTION- Data compression request for Output: SYST (syst/copy/yes/no)
CAT - Catalog Output tape : NO (yes/no/only/recat)
Display more FATAR Operands?: NO (yes/no)

Option 5: Logical File Copy

Option 5 on the FATAR Selection Menu brings you to this panel, which uses FATAR to logically copy files from an input tape (or multi-volume tape set) to an output tape (or tape set). If you use the default of LABEL=SL (or AL) on TAPEIN, the actual volume serials of the input tape(s) and the actual data set name of the **first** data set on the first volume must be provided. If you change the TAPEIN to LABEL=BLP, any dsname and volume serial(s) can be used; FATAR will recognize the label type of the input tape automatically even with BLP. For TAPEOUT, the LABEL parameter determines the format of the output tape (SL for IBM standard labels, AL for ANSI labels, NL or BLP for unlabeled tape); if the output tape is labeled, FATAR will extract the data set names of the files on the input tape and change the name of the output files to be the same; the data set name of TAPEOUT is not used. DCB information can be omitted if the input tape has labels; FATAR will get the file characteristics from those labels. If the full data set name is available (see Section 21), you can specify YES or RECAT for CAT to catalog the data set to the output tape (use RECAT if the input data sets are already cataloged).

RECOVER DATA FROM AN OVERWRITTEN TAPE


V4.9.xx ---------- FATAR OVER-WRITTEN TAPE RECOVERY ----------
COMMAND ===>
Enter QI or QO for RMM TMS Query
TAPEIN DD Dsname=
Unit =(TAPE),Label=(@FZLABLI)
VolSer=(99999G)
Dcb =(
TAPEOUT DD Dsname=
Unit =(TAPE),Label=(@FZLABLO)
VolSer=(
Dcb =(
NUMFILES - Number of Logical files to be read.....: 0 (zero = all files)
OUTBLP - By-pass label processing on Output.....: NO (no/yes)
VALIDATE - Verify output Volser/Expiration Date...: NO (no/opt)
VERIFY - Verify readability of Output tape.....: NO (no/yes)
EODFIND - Recover all files past 1st EOD mark....: YES (no/yes)
RECUSERL - Recover as if they contain user labels.. NO (no/yes)
EODRETRY - Nbr of attempts to position past EOD...:
COMPACTION- Data compression request for Output... : SYST (syst/copy/no/yes)
Display more FATAR Operands?: NO (no/yes)

Option 6: Recover Data

Option 6 on the FATAR Selection Menu brings you to this panel, which uses FATAR to recover data from a tape that has been overwritten with a smaller amount of data; the data beyond the overwrite can be recovered. The parameters are similar to Option 5 (COPY). However, BLP is required to position the tape to the beginning of the overwritten data so the submitted job must be authorized to use BLP. A label number **must** be specified, pointing immediately after the overwriting data; since a single labeled data set uses 4 physical files, LABEL=5 is usually appropriate (see Section 24.2 for more guidance on recovering overwritten data).

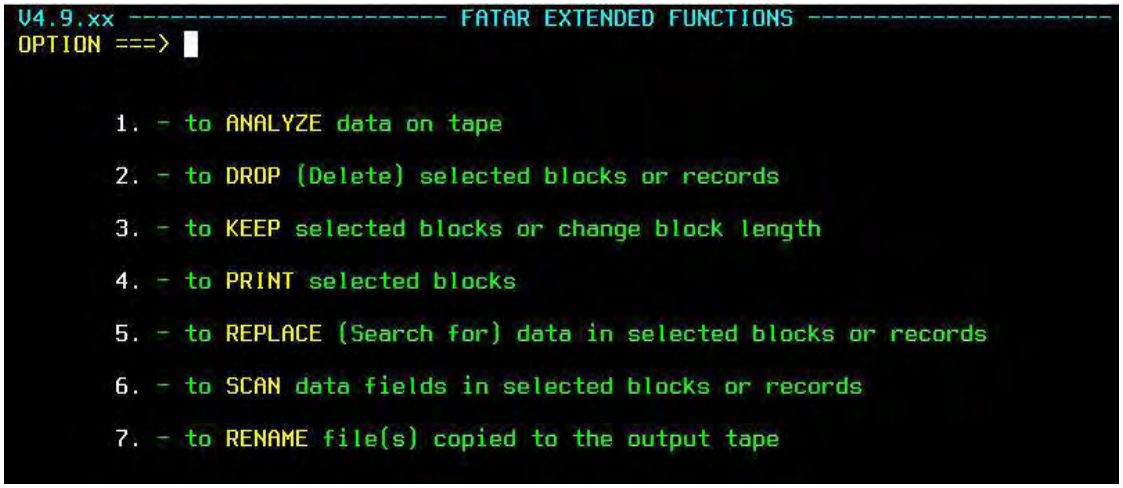
For cartridge tapes that write an end-of-data mark following the portion of the tape currently in use, such as the IBM 3490E, IBM 3590, IBM 3952, STK/SUN 9840, STK/SUN 9940, and StorageTek T10000B, the above instructions don't apply. If you wish to recover files from a tape that contains an end-of-data mark, you need to code 'YES' in the EODFIND field. You must also specify the DSN of the overwritten data set in the DSNAME field on **both** TAPEIN and TAPEOUT, code the DCB information for the overwritten data set in the DCB field for **both** TAPEIN and TAPEOUT, and change the Label= for TAPEOUT **only** to LABEL=(,SL,EXPDT=98000).

Caution: Recovering files from an overwritten tape that contains an end-of-data mark requires you to follow the above instructions exactly, or the output tape may be unusable. If you have any doubts about the correct method to use to recover a tape, call Innovation for assistance.

93.3 FATAR EXTENDED FUNCTIONS ISPF INTERFACE

FATAR EXTENDED FUNCTIONS

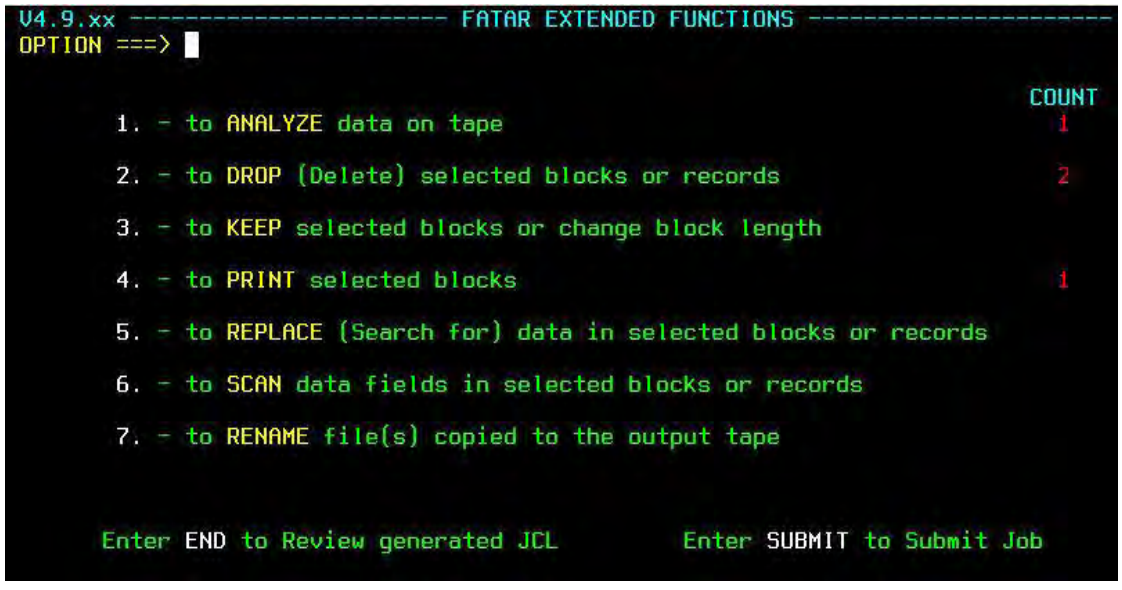
Option 7 (FATAR Extended Functions) on the FATAR Selection Menu brings you to this panel, which is a starting menu for an interactive dialog that can build a FATAR job stream using the many powerful features of FATAR. You should familiarize yourself with the FATAR Modification/Scan control statements and their functions (Section 23.3) before attempting to use these panels.


V4.9.xx FATAR EXTENDED FUNCTIONS
OPTION ===>
1. - to ANALYZE data on tape
2. - to DROP (Delete) selected blocks or records
3. - to KEEP selected blocks or change block length
4. - to PRINT selected blocks
5. - to REPLACE (Search for) data in selected blocks or records
6. - to SCAN data fields in selected blocks or records
7. - to RENAME file(s) copied to the output tape

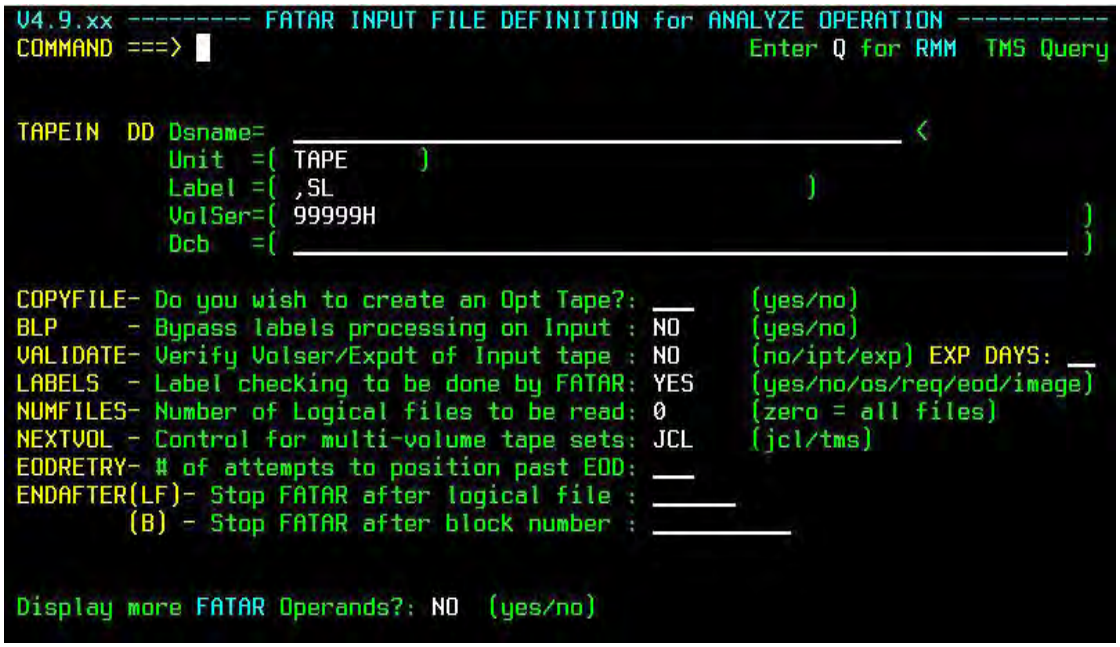
FATAR Extended Functions Menu

Option 1 can be invoked one time to specify the input and optional output tapes and generate the JCL . If you invoke any of the other options before Option 1, it will take you through Option 1 first. The other options, 2 through 6, can be invoked repeatedly to generate whatever set of FATAR functions are required in that FATAR step.

As you invoke the various options, this menu will be updated to show how many statements of each kind have been generated. When complete, you can submit the generated job directly, or enter ISPF EDIT to review and submit or save it.


V4.9.xx FATAR EXTENDED FUNCTIONS
OPTION ===>
COUNT
1. - to ANALYZE data on tape 1
2. - to DROP (Delete) selected blocks or records 2
3. - to KEEP selected blocks or change block length
4. - to PRINT selected blocks 1
5. - to REPLACE (Search for) data in selected blocks or records
6. - to SCAN data fields in selected blocks or records
7. - to RENAME file(s) copied to the output tape
Enter END to Review generated JCL
Enter SUBMIT to Submit Job

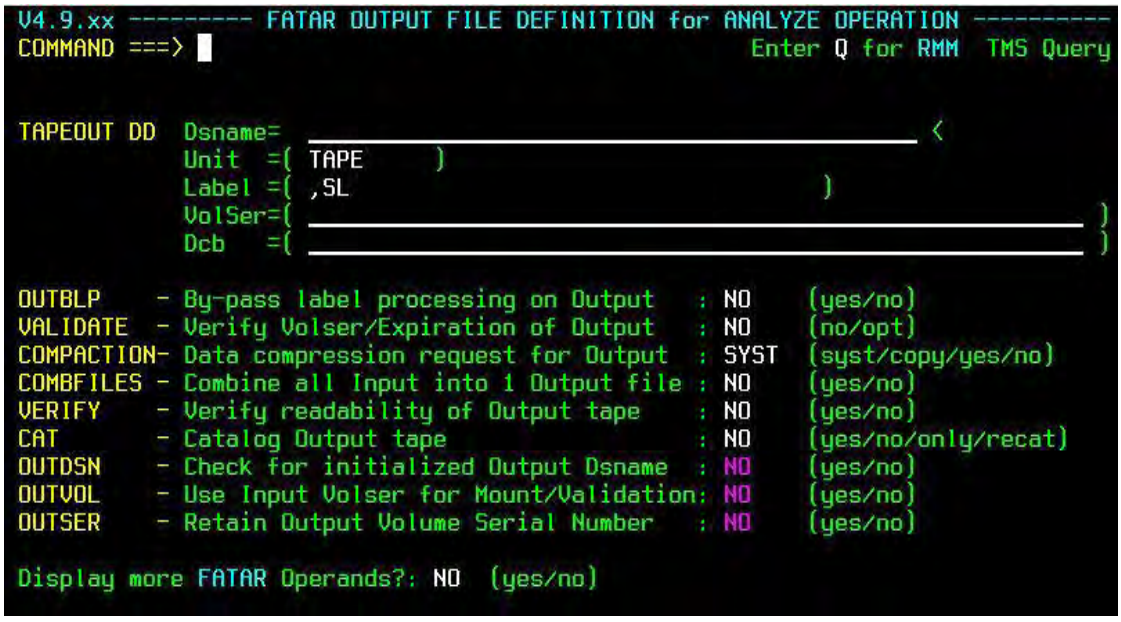
TAPEIN AND TAPEOUT SPECIFICATIONS


V4.9.xx --------- FATAR INPUT FILE DEFINITION for ANALYZE OPERATION ----------
COMMAND ===>
Enter Q for RMM TMS Query
TAPEIN DD Dsname=
Unit =(TAPE)
Label =(,SL)
VolSer=(99999H)
Dcb =( )
COPYFILE- Do you wish to create an Opt Tape?: (yes/no)
BLP - Bypass labels processing on Input : NO (yes/no)
VALIDATE- Verify Volser/Expdt of Input tape : NO (no/ipt/exp) EXP DAYS:
LABELS - Label checking to be done by FATAR: YES (yes/no/os/req/eod/image)
NUMFILES- Number of Logical files to be read: 0 (zero = all files)
NEXTVOL - Control for multi-volume tape sets: JCL (jcl/tms)
EODRETRY- # of attempts to position past EOD:
ENDAFTER(LF)- Stop FATAR after logical file :
(B) - Stop FATAR after block number :
Display more FATAR Operands?: NO (yes/no)

Extended Functions Option 1

Option 1 will take you to this screen to specify the input tape; you also come here if you enter another option without using Option 1 first. Refer to page 93-18 for considerations for the input (and output) tape specification.

If you specify YES for COPYFILE, the next panel allows you to specify the output tape, including the ability to optionally catalog the output files.


V4.9.xx --------- FATAR OUTPUT FILE DEFINITION for ANALYZE OPERATION ----------
COMMAND ===>
Enter Q for RMM TMS Query
TAPEOUT DD Dsname=
Unit =(TAPE)
Label =(,SL)
VolSer=( )
Dcb =( )
OUTBLP - By-pass label processing on Output : NO (yes/no)
VALIDATE - Verify Volser/Expiration of Output : NO (no/opt)
COMPACTION- Data compression request for Output : SYST (syst/copy/yes/no)
COMBFILES - Combine all Input into 1 Output file : NO (yes/no)
VERIFY - Verify readability of Output tape : NO (yes/no)
CAT - Catalog Output tape : NO (yes/no/only/recat)
OUTDSN - Check for initialized Output Dsname : NO (yes/no)
OUTVOL - Use Input Volser for Mount/Validation: NO (yes/no)
OUTSER - Retain Output Volume Serial Number : NO (yes/no)
Display more FATAR Operands?: NO (yes/no)

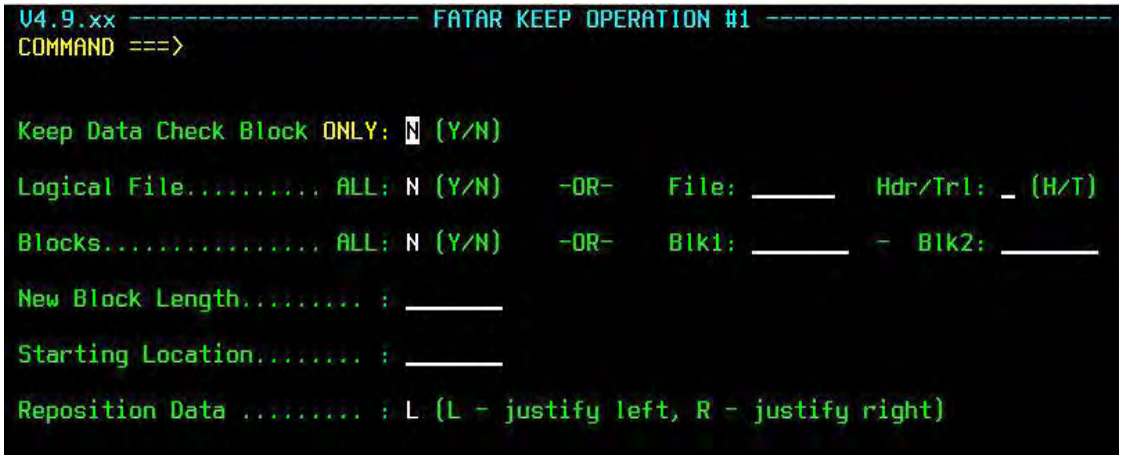
FATAR DROP OPERATION

V4.9.xx ---------- FATAR DROP OPERATION #1 ----------
COMMAND ===>
Drop Data Check Block ONLY: N (Y/N)
Logical File..... ALL: N (Y/N) -OR- File: _____ Hdr/Trl: _ (H/T)
Blocks/Records..... ALL: N (Y/N) -OR- Blk1: _____ . Rcd1: _____
Blk2: _____ . Rcd2: _____

Extended Functions Option 2

This is Option 2 on the FATAR Extended Functions Panel, which allows you to drop whole blocks or logical records from the output tape. See Section 23.3.2 for guidance on the DROP statement.

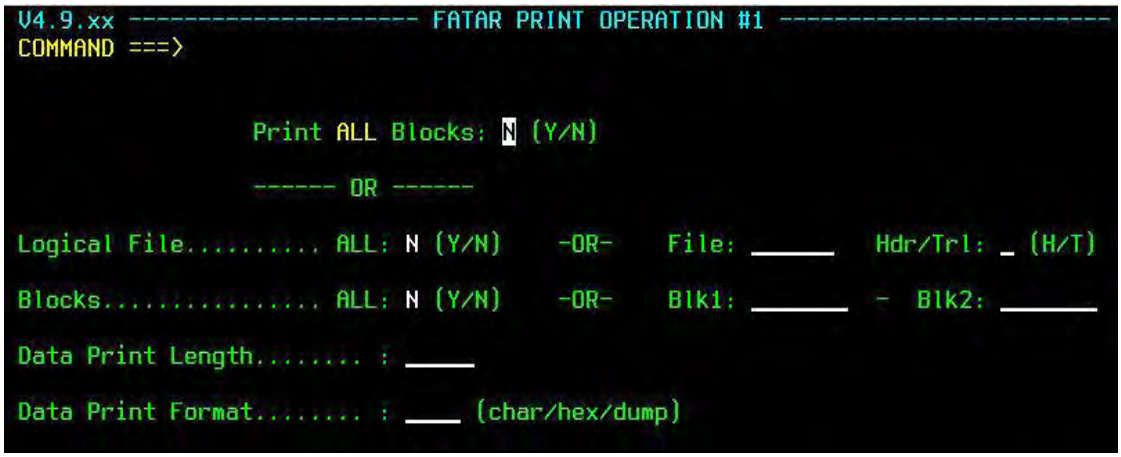
FATAR KEEP OPERATION


V4.9.xx ---------- FATAR KEEP OPERATION #1 ----------
COMMAND ===>
Keep Data Check Block ONLY: N (Y/N)
Logical File..... ALL: N (Y/N) -OR- File: _____ Hdr/Trl: _ (H/T)
Blocks..... ALL: N (Y/N) -OR- Blk1: _____ - Blk2: _____
New Block Length..... : _____
Starting Location..... : _____
Reposition Data ..... : L (L - justify left, R - justify right)

Extended Functions Option 3

This is Option 3 on the FATAR Extended Functions Panel, which allows you to copy whole blocks or logical records to the output tape when they would otherwise be dropped. It can also modify the length of copied blocks. See Section 23.3.3 for guidance on the KEEP statement.

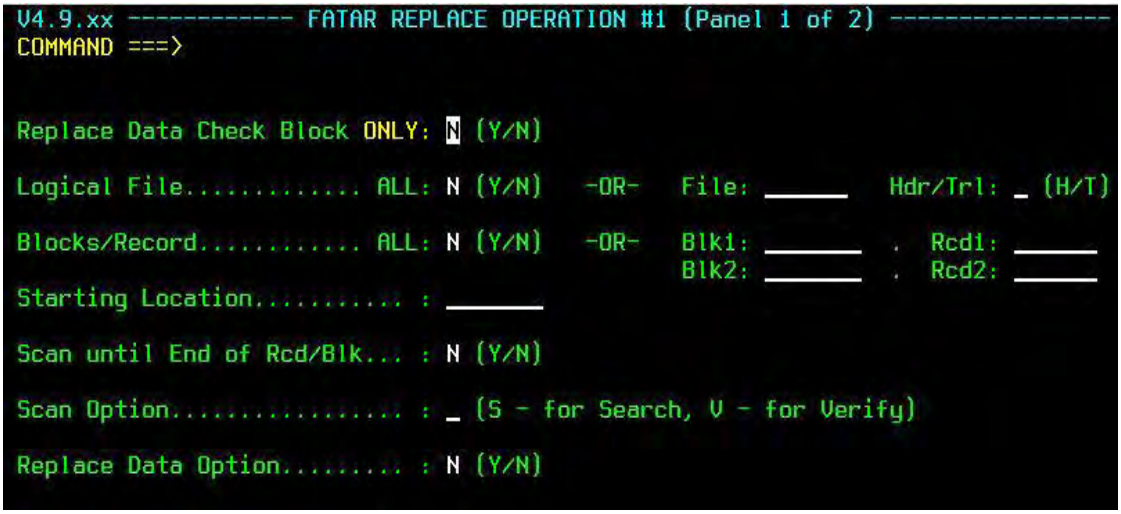
FATAR PRINT OPERATION


V4.9.xx ---------- FATAR PRINT OPERATION #1 ----------
COMMAND ===>
Print ALL Blocks: N (Y/N)
------ OR ------
Logical File..... ALL: N (Y/N) -OR- File: _____ Hdr/Trl: _ (H/T)
Blocks..... ALL: N (Y/N) -OR- Blk1: _____ - Blk2: _____
Data Print Length..... : ____
Data Print Format..... : ___ (char/hex/dump)

Extended Functions Option 4

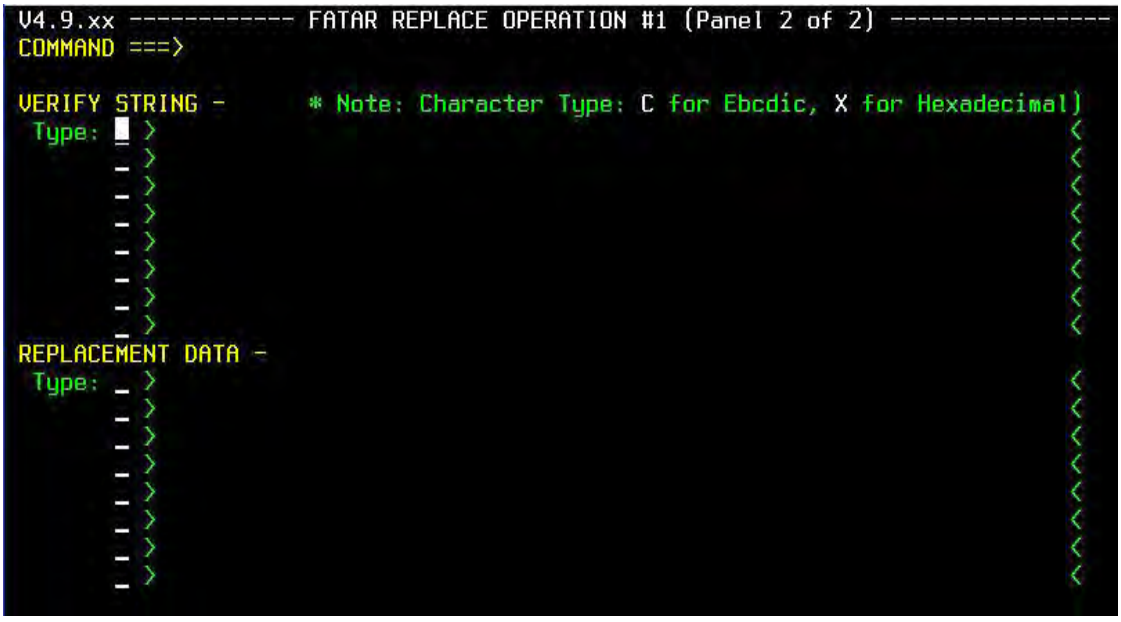
This is Option 4 on the FATAR Extended Function Panel, which allows you to print blocks or records from the input tape. See Section 23.3.1 for guidance on the PRINT statement.

FATAR REPLACE OPERATION


V4.9.xx ---------- FATAR REPLACE OPERATION #1 (Panel 1 of 2) ----------
COMMAND ===>
Replace Data Check Block ONLY: N (Y/N)
Logical File..... ALL: N (Y/N) -OR- File: _____ Hdr/Trl: _ (H/T)
Blocks/Record..... ALL: N (Y/N) -OR- Blk1: _____ . Rcd1: _____
Blk2: _____ . Rcd2: _____
Starting Location..... : _____
Scan until End of Rcd/Blk... : N (Y/N)
Scan Option..... : _ (S - for Search, V - for Verify)
Replace Data Option..... : N (Y/N)

Extended Functions Option 5, panel 1

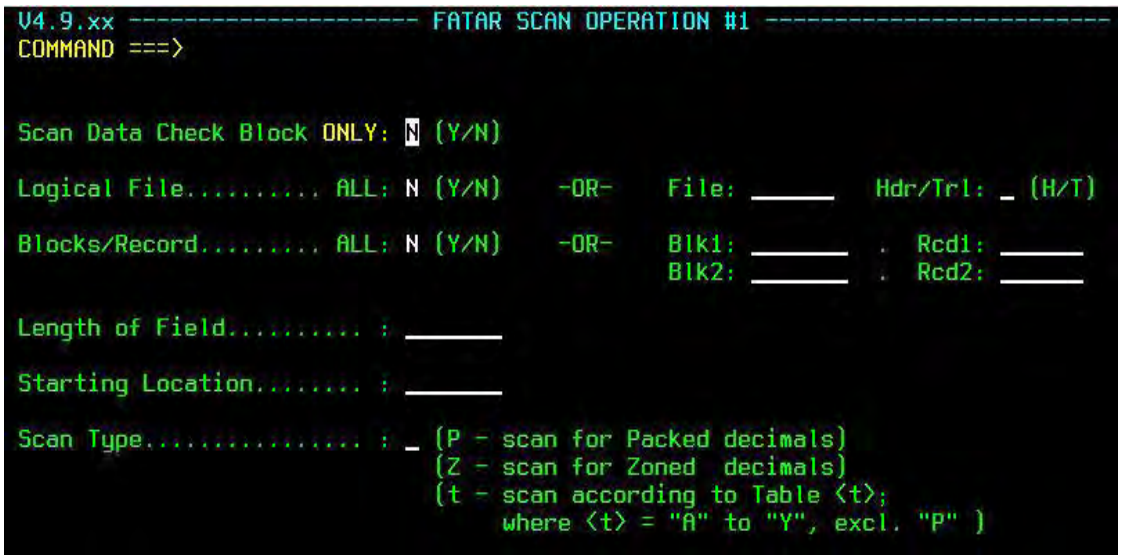
This is Option 5 on the FATAR Extended Function Panel, which allows you to scan for data in blocks or records on the input tape and optionally replace data in blocks or records written to the input tape. See Section 23.3.4 for guidance on the REPLACE statement.


V4.9.xx ---------- FATAR REPLACE OPERATION #1 (Panel 2 of 2) ----------
COMMAND ===>
VERIFY STRING - * Note: Character Type: C for Ebcdic, X for Hexadecimal
Type: _ >
REPLACEMENT DATA -
Type: _ >

Extended Functions Option 5, panel 2

This is the second panel for the REPLACE function, where the data to be located and the replacement data are entered as either character or hexadecimal strings. The panel may contain either the SEARCH STRING field, the REPLACEMENT DATA field, or both, depending on options entered on the first panel.

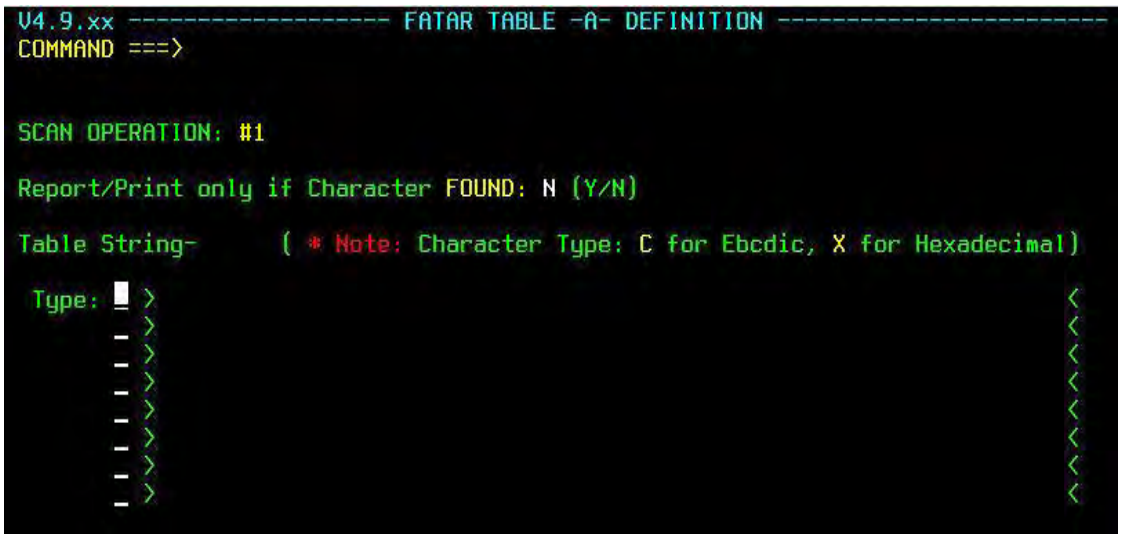
FATAR SCAN OPERATION


V4.9.xx ---------- FATAR SCAN OPERATION #1 ----------
COMMAND ===>
Scan Data Check Block ONLY: N (Y/N)
Logical File..... ALL: N (Y/N) -OR- File: _____ Hdr/Trl: _ (H/T)
Blocks/Record..... ALL: N (Y/N) -OR- Blk1: _____ . Rcd1: _____
Blk2: _____ . Rcd2: _____
Length of Field..... : _____
Starting Location..... : _____
Scan Type..... : _ (P - scan for Packed decimals)
(Z - scan for Zoned decimals)
(t - scan according to Table <t>;
where <t> = "A" to "Y", excl. "P")

Extended Functions Option 6

This is Option 6 on the FATAR Extended Function Panel, which allows you to validate data on the input tape; you can check for valid packed or zoned decimal fields, or validate against a table of valid bytes you provide. See Section 23.3.5 for guidance on the SCAN statement.

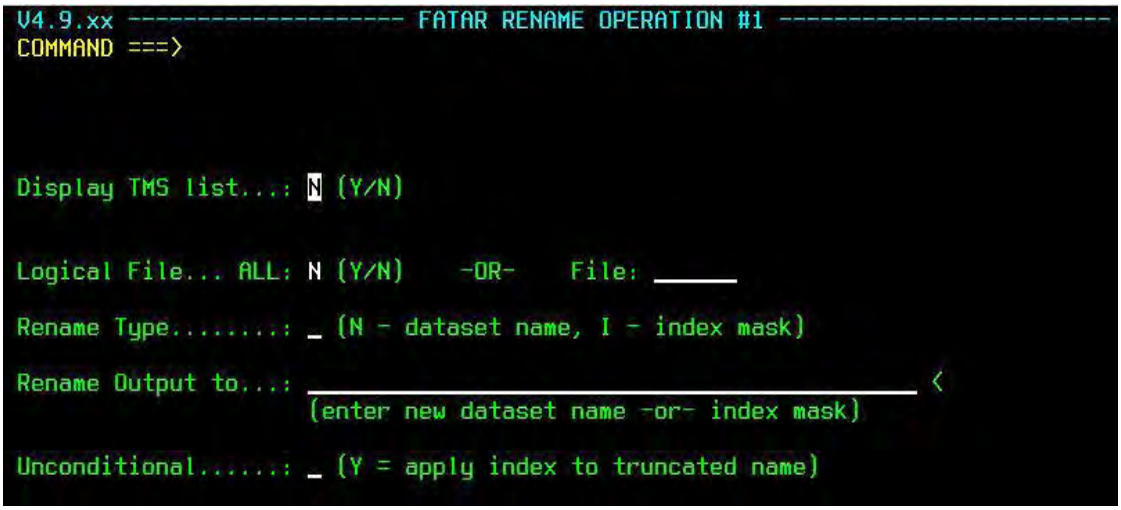
FATAR TABLE DEFINITION


V4.9.xx ---------- FATAR TABLE -A- DEFINITION ----------
COMMAND ===>
SCAN OPERATION: #1
Report/Print only if Character FOUND: N (Y/N)
Table String- (* Note: Character Type: C for Ebcdic, X for Hexadecimal)
Type: _ >

Scan Table Definition panel

If you specify a Table Name other than P or Z on the previous panel, this panel allows you to specify the characters that are valid (or invalid) for the field to be validated. See Section 23.3.6 for guidance on the TABLE statement

FATAR RENAME STATEMENT


V4.9.xx ---------- FATAR RENAME OPERATION #1 ----------
COMMAND ===>
Display TMS list...: N (Y/N)
Logical File... ALL: N (Y/N) -OR- File: _____
Rename Type.....: _ (N - dataset name, I - index mask)
Rename Output to...: _____ <
(enter new dataset name -or- index mask)
Unconditional.....: _ (Y = apply index to truncated name)

Extended Functions Option 7

This is Option 7 on the FATAR Extended Function Panel, which allows you to rename output data sets (if you specified an output tape); you can check for valid packed or zoned decimal fields, or validate against a table of valid bytes you provide. See Section 23.3.7 for guidance on the RENAME statement.

NOTES

VIEWING/PRINTING THIS MANUAL

A copy of this manual is provided on a CD-ROM in Adobe Acrobat PDF format, along with a copy of the Adobe Acrobat Reader, enabling you to view and print the manual in a format identical to the printed manual. It is also available for downloading at the Innovation FTP web site: http://www.innovationdp.fdr.com/ftp/ftp.cfm

READER'S COMMENT FORM

FATS/FATAR/FATSCOPY USER'S MANUAL V4.9.28

INNOVATION DATA PROCESSING

If you have comments on this product or its manual, including:

- errors in the text or typographical errors;
- clarity;
- suggestions for improvement in the manual;
- suggestions for improvement in the product; or
- any other comments,

please complete this form and fax it to Innovation at 973-890-7147 (in Europe you may fax it to your local Innovation office as shown on the front page of the manual). You may also e-mail your comments to Innovation at support@fdrinnovation.com; be sure to identify the manual name and version number in your message.

Your name: _____

Company name: _____

Mailing Address: _____

_____

_____

E-mail address: _____

Comments: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____